



AdminStudio 9.5

User Guide

AdminStudio 9.5 User Guide

Part Number: ADS-0950-UG01
Product Release Date: 11/23/2009
Document Revision Date: 02/16/2010

Copyright Notice

Copyright © 2001–2010 Flexera Software, Inc. and/or InstallShield Co. Inc. All Rights Reserved.

This product contains proprietary and confidential technology, information and creative works owned by Flexera Software, Inc. and/or InstallShield Co. Inc. and their respective licensors, if any. Any use, copying, publication, distribution, display, modification, or transmission of such technology in whole or in part in any form or by any means without the prior express written permission of Flexera Software, Inc. and/or InstallShield Co. Inc. is strictly prohibited. Except where expressly provided by Flexera Software, Inc. and/or InstallShield Co. Inc. in writing, possession of this technology shall not be construed to confer any license or rights under any Flexera Software, Inc. and/or InstallShield Co. Inc. intellectual property rights, whether by estoppel, implication, or otherwise.

All copies of the technology and related information, if allowed by Flexera Software, Inc. and/or InstallShield Co. Inc., must display this notice of copyright and ownership in full.

Trademarks

Flexera Software, AdminStudio, FLEXenabled, FLEXIm, FLEXnet, FLEXcertified, FLEXnet Connect, FLEXnet Connector, FLEXnet Manager, FLEXnet Publisher, Globetrotter, InstallShield, InstallShield Developer, InstallShield DevStudio, InstallShield Professional, and Package For The Web are registered trademarks or trademarks of Flexera Software, Inc. and/or InstallShield Co. Inc. in the United States of America and/or other countries. All other brand and product names mentioned herein are the trademarks and registered trademarks of their respective owners.

Restricted Rights Legend

The software and documentation are “commercial items,” as that term is defined at 48 C.F.R. §2.101, consisting of “commercial computer software” and “commercial computer software documentation,” as such terms are used in 48 C.F.R. §12.212 or 48 C.F.R. §227.2702, as applicable. Consistent with 48 C.F.R. §12.212 or 48 C.F.R. §227.2702-1 through 227.2702-4, as applicable, the commercial computer software and commercial computer software documentation are being licensed to U.S. government end users (A) only as commercial items and (B) with only those rights as are granted to all other end users pursuant to the terms and conditions set forth in the Flexera Software, Inc. standard commercial agreement for this software. Unpublished rights reserved under the copyright laws of the United States of America.

Contents

Part 1: Getting Started	51
1 AdminStudio 9.5 Help Library	53
What's New in AdminStudio 9.5	57
AdminStudio 9.5 SP1	57
Repackaging 64-Bit Applications	57
Creating 64-Bit Microsoft App-V Applications	57
PackageExpert Windows 7 Compatibility Tests	57
AdminStudio 9.5	58
AdminStudio Editions	70
Activating AdminStudio	73
Evaluating AdminStudio	74
Upgrading Your Product Edition	76
Using Help	77
Contacting Us	79
2 Using the Process Assistant	81
Using the Process Assistant	82
Home Page	83
Package Page	84
Customize Page	84
Adding Transforms to an MSI Package	85
Resolve Conflicts Page	85
Importing Applications into the Application Catalog	86
Application Conflicts	86
Automatic and Manual Conflict Resolution	86

ConflictSolver Process Assistant	.87
Select a Package Page	.88
Current and New Packages	.89
Draft Packages	.90
Creating a New Group	.91
Configuring Import Options	.91
Patch Files	.91
Extended Attributes Page	.92
Validation Page	.92
Microsoft Internal Consistency Evaluators (ICEs)	.93
Validation Error Severity Ratings	.93
Configuring Validation Options	.94
Best Practices Page	.95
Best Practice ACEs	.95
Resolving Best Practice ACE Errors	.99
Configuring Conflict Detection	.100
Configuring Resolution Options	.100
Detect Conflicts Page	.101
Target Packages	.101
Configuring Conflict Detection	.101
Resolve Conflicts Page	.102
Automatic and Manual Conflict Resolution	.102
Configuring Resolution Options	.103
Reporting Page	.104
Organize Package Page	.105
Current and New Packages	.106
Draft Packages	.107
Creating a New Group	.108
Test Page	.108
Distribute Page	.109
Distributing Transforms With an MSI Package	.110

3 Using the AdminStudio Interface 111

Configuring the AdminStudio Interface	.112
Launching AdminStudio Applications	.112
Specifying the Shared AdminStudio Application Catalog	.114
Specifying the AdminStudio Shared Location	.114
Setting the Workflow Task Help Page Location	.115
Configuring How Often AdminStudio Checks for Updates	.115
Configuring AdminStudio to Stay on Top	.116
Generating a Debug Log for AdminStudio	.116
Specifying the AdminStudio Enterprise Server URL	.116
Specifying the Configuration Manager Web Console URL	.117

Working with Tools	117
Adding New Tools to the Tools Gallery	118
Editing Properties for an Existing Tool	118
Adding Command-Line Configurations for an Existing Tool	119
Modifying Command-Line Configurations for an Existing Tool	119
Deleting Command-Line Configurations from an Existing Tool	120
Associating Tools with Tasks	120
Running Associated Tools in Projects	121
Deleting Existing Tools	121
Limiting Tool Accessibility	122
Workflows and Projects	122
Creating and Editing Workflows	122
Creating New Workflows	123
Renaming Workflows	123
Filtering Workflows	123
Deleting Workflows	124
Creating New Tasks	124
Modifying Task Properties	124
Creating Notes for a Task	125
Renaming Tasks	125
Reordering Tasks	125
Associating Help Files with Tasks	126
Deleting Tasks	126
Adding New Tools from the Workflow Templates Tab	126
Creating and Using Projects	127
Creating Workflows with the New Workflow Project Wizard	127
Filtering Projects	128
Executing Projects	128
Running Associated Tools in Projects	128
Deleting Projects	129
Saving Workflow and Project Changes	129
Workflow Project Example: Using the New Workflow Project Wizard	129
Workflows, Projects, and Permissions	131
Frequently Asked Questions	132
AdminStudio Interface Reference	133
AdminStudio Start Page	134
Assistant Tab	136
Tools Tab	137
Workflow Templates Tab	138
Tasks	138
Projects Tab	139
Enterprise Server Tab	139
Workflow Manager Tab	140
AdminStudio Menus and Toolbar	140

Contents

Dialog Boxes	143
About AdminStudio Dialog Box	143
Add New Tool Dialog Box	144
Command Line Properties Dialog Box	145
Options Dialog Box	145
Application Catalog Tab	146
Locations Tab	146
Updates Tab	147
Virtual Packages Tab	147
Quality Tab: Customer Experience Improvement Program	148
Tool Properties Dialog Box	149
Properties Tab	149
Configuration Tab	150
Wizards	152
Add Tool Wizard	152
Welcome Panel	152
Tool Properties Panel	152
Command-Line Configurations Panel	153
New Workflow Project Wizard	153
Welcome Panel	153
Workflow Selection Panel	154
Source Package Panel	154
Target Directory and File Name Panel	154
Log Files	155

Part 2: Managing Users, Roles, Permissions, Application Catalogs. . 157

4 Managing Users, Directory Services, and User Logins. 159

Managing Users and Groups	159
Creating a New User Account	160
Importing Directory Services Users and Groups	160
Disabling a User or Group Account	161
Deleting a User Account	162
Managing Directory Services Configurations	163
Managing Directory Services Connections	163
Adding a Directory Service Connection	163
Editing an Existing Directory Service Connection	166
Deleting a Directory Service Connection	167
Managing Directory Services Attributes	167
Managing User Logins.	168
Login Methods	169
IIS Authentication Methods Dialog Box	170
Using AdminStudio Enterprise Server Account Login	171

Using Domain Account Login	171
Using Single Sign-On Login	172
Using Guest Account Login	173
Setting Up a Guest Account	173
Logging in as a Guest	174
Setting the Session Timeout Value	175
Users/Groups and Directory Services Reference	176
User Administration Page	176
User Details View	178
Directory Services User/Group Add View	179
Directory Services Page	180
Directory Services List Page	180
Add Directory Service Connection View	181
Directory Services Attributes Administration Page	183
Add Directory Service Attributes View	184
5 Managing Roles and Permissions	185
AdminStudio and Workflow Manager Roles and Permissions	185
Role Permission Lists	186
AdminStudio Client Tools Permissions	186
AdminStudio Enterprise Server Tools Permissions	189
Workflow Manager Permissions	191
System Roles	194
Superuser Role: AMSSuper	194
Default System Roles	195
Default System Users	196
Creating a New Role	197
Editing an Existing User Role	197
Copying an Existing Role	198
Deleting a Role	198
Roles Reference	199
Role Administration Page	199
Copy Role View	200
Role Details View	201
6 Managing Application Catalog Databases	203
Application Manager and ConflictSolver	204
Functionality Comparison	205
Accessing the Tools	206
Displaying ConflictSolver Options in Application Manager	207
Creating and Connecting to Application Catalogs in Application Manager and ConflictSolver	207
Managing Application Catalogs	207
About AdminStudio Application Catalogs	208

Contents

Overview of Application Catalogs	208
Standalone Application Catalog vs. the AdminStudio Enterprise Server Application Catalog	209
Using the AdminStudio Quick Start Evaluation Experience	210
Connecting to an Existing Application Catalog	210
Connecting AdminStudio Client Tools to a Standalone Application Catalog	211
Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog	212
Creating New Application Catalogs	213
Creating New Application Catalogs Using the AdminStudio Interface	213
Creating New Application Catalogs Using Scripts	214
Specifying a Default AdminStudio Application Catalog	216
Using the AdminStudio Interface	217
Editing the Shared AdminStudio.ini File	218
Searching an Application Catalog	219
Disconnecting from an Application Catalog	220
Upgrading an Existing Application Catalog	221
Organizing Your Application Catalog Using Groups	221
Adding Groups	222
Organizing Products in Application Manager	222
Deleting Application Manager Groups	223
Editing Group Properties	223
Copying Packages to Multiple Groups	223
Moving Products, OS Snapshots, and Groups	224
Viewing and Editing Package Extended Attributes	224
Using Extended Attributes	225
Extended Attribute Description File	225
Integrating Package Extended Attribute Data with an Application Request	227
Viewing Package History	228
Deleting Packages	230
Importing Data	230
Importing Windows Installer Packages, Transforms, and Patches	231
Importing Windows Installer Packages	232
About Windows Installer Packages (.msi)	234
Using Duplicate Package Identifiers	234
Performing an Ad-Hoc Import of Transform Files or Patch Files	236
About Transforms (.mst)	238
About Patches (.msp)	238
Importing Virtual Packages	238
Importing a Virtual Package During the Import of its Source Windows Installer Package	239
Importing a Virtual Package After the Import of its Source Windows Installer Package	241
Importing a Virtual Package Without Its Source Windows Installer Package	243
Manually Associating a Virtual Package with a Windows Installer Package	243
Deleting a Virtual Package Association	244
Importing Merge Modules	244
About Merge Modules (.msm)	245

Importing OS Snapshots	245
About OS Snapshots (.osc)	246
Importing Marimba NCP Files	247
About Marimba Native Channel Packager Files (.ncp)	248
Importing Other Setup Types	249
About Other Setup Types	250
Using the Software Repository	250
Enabling the Software Repository in Application Catalogs	251
Enabling the Software Repository in a New Application Catalog	252
Enabling the Software Repository in an Existing Application Catalog	252
Methods to Import Packages into the Software Repository	253
Importing a Package into the Software Repository Using the Import Wizard	253
Adding a Package to the Software Repository Using the ConflictSolver Process Assistant	254
Adding a Package to the Software Repository by Overwriting Original Package	254
Software Repository Auto Import Method	255
Identifying Software Repository Packages in Application Manager	255
Using Version Management Features	256
Checking-Out and Checking-In Packages	257
Cancel Check Out	257
Getting a Copy of the Latest Version of a Package	258
Software Repository Integration into Other AdminStudio Tools	258
Sharing Application Catalog Data	260
Merging Application Catalogs Using the Merge Wizard	264
Automatically Importing Packages	265
Automatically Importing Packages in a Remote Application Catalog	266
Linking to a Remote Application Catalog	266
Editing a Link to a Remote Application Catalog	269
Manually Updating a Linked Package	270
Deleting a Link to a Remote Application Catalog	271
Viewing Remote Package Links on the Application Manager Product View	271
Automatically Importing Packages from a Network Directory	272
Linking to a Network Directory	272
Editing a Link to a Network Directory	274
Forcing an Import of Packages in a Network Directory	275
Deleting a Link to a Network Directory	275
Package Auto Import and Duplicate Package Names	275
Replicating Application Catalogs	276
About Application Catalog Replication	277
Benefits of Using Application Catalog Replication	277
Creating and Managing Publisher and Subscriber Application Catalogs	278
Configuring the AdminStudio Shared Location for Replication	280
Initial Configuration Checklist	280
Publication/Subscription Lifecycle	281
Specialized User Scenarios	283

Creating and Managing Publications	284
Managing Publications	285
Creating a New Publication	286
Publishing a Publication	288
Setting a Publication Schedule	289
Editing a Publication	289
Editing a Publication Access List	290
Deleting a Publication	290
Creating and Managing Subscriptions	291
Managing Subscriptions	291
Creating a New Subscription	292
Enabling/Disabling a Subscription	294
Manually Updating a Subscription	295
Deleting a Subscription	295
Using Microsoft SQL Server to Perform Merge Replication of Application Catalogs	296
Taking OS Snapshots	302
OS Snapshot Best Practices	302
Configuring OS Snapshot Analysis Options	303
Capturing an OS Snapshot	304
Reference	304
Application Manager Interface	305
Menus and Toolbar	305
Product View Icons	308
Merge Modules View Icons	311
Context Menus	312
Output Window	316
User Permissions in Application Manager	317
Application Manager Views	318
Application Catalog View	318
Group View	319
Product View	319
Extended Attributes View	321
Files/Components View	322
INI File Changes View	323
Registry View	323
Shortcuts View	324
Merge Modules View	324
History View	325
Tables View	325
OS Snapshot View	326
Extended Attributes View for OS Snapshots	326
Files View for OS Snapshots	326
INI File Changes View for OS Snapshots	327
Registry View for OS Snapshots	327

Shortcuts View for OS Snapshots	328
Tables View for OS Snapshots	328
Other Setup Types View	328
Extended Attributes View for Other Setup Types	329
Marimba NCP Files Views	329
Extended Attributes View for Marimba NCP Files	330
Files View for Marimba NCP Files	330
INI File Changes View for Marimba NCP Files	330
Registry View for Marimba NCP Files	331
Shortcuts View for Marimba NCP Files	332
Tables View for Marimba NCP Files	332
Virtual Package View	332
Patches Tab Views	334
Patches Group View	334
New Patches Group View	335
Group View of a Selected Group	336
Patch View	336
Merge Module Views	338
All Merge Modules View	338
Merge Module View	338
Merge Module / Components View	339
Merge Module / Dependency View	339
Merge Module / Exclusion View	339
Merge Module / Files View	340
Merge Module / Products View	340
Dialog Boxes	340
Application Catalog Properties Dialog Box	341
Associate with Workflow Manager Application Dialog Box	342
Change Enterprise Server Password Dialog Box	342
Command-Line Parameters Dialog Box	343
Connect Application Catalog Dialog Box	343
Enterprise Server Tab	344
Standalone Tab	345
Recent Tab	346
Default Application Catalog Dialog Box	347
Extended Attribute Property Dialog Box	348
Find Dialog Box	348
Login Required Dialog Box	350
Group Properties Dialog Box	350
Package Auto Import Dialog Box	351
Package Auto Import Properties (Remote Application Catalog) Dialog Box	352
Package Auto Import Properties (Network Directory) Dialog Box	354
Publication Manager Dialog Box	355
Publication Processing Dialog Box	356

Publication Properties Dialog Box	356
Select Application Catalog Dialog Box	357
Select AdminStudio Enterprise Server URL Dialog Box	357
Subscription Manager Dialog Box	357
Task Scheduling Dialog Box	359
<i>Test Connection Dialog Box</i>	360
Virtual Package Association Dialog Box	360
Wizards	361
Application Catalog Wizard	361
<i>Welcome Panel</i>	361
<i>Specify Database Information Panel</i>	362
<i>Select Software Repository Location Panel</i>	363
<i>Creating Application Catalog Panel</i>	364
Import Wizard	364
<i>Welcome Panel</i>	365
<i>File Selection Panel</i>	366
<i>Associate Virtual to MSI Packages Panel</i>	367
<i>MST Source Information Panel</i>	368
<i>MSP Source Information Panel</i>	368
<i>MSM Source Information Panel</i>	369
<i>Virtual Package Import Panel</i>	370
<i>Target Package Information Panel</i>	371
<i>Target Package Information Panel (Software Repository)</i>	371
<i>Target Package Information Panel (Transforms/Patches)</i>	372
<i>Import Options Panel</i>	372
<i>Additional non-MSI Import Options Panel</i>	372
<i>Destination Group Panel</i>	373
<i>Summary Panel</i>	373
Package Auto Import Wizard	373
<i>Welcome Panel</i>	374
<i>Package Auto Import Type Selection Panel</i>	374
<i>Remote Link Application Catalog Panel</i>	374
<i>Folder Panel</i>	375
<i>Groups Panel</i>	375
<i>Destination Group Panel</i>	375
<i>Schedule Panel</i>	376
<i>Summary Panel</i>	376
Merge Wizard	376
<i>Welcome Panel</i>	377
<i>Source Application Catalog Panel</i>	377
<i>Groups Panel</i>	377
<i>Progress Panel</i>	377
OS Snapshot Wizard	378
<i>Welcome Panel</i>	378

<i>Project Information Panel</i>	378
<i>Analyzing Panel</i>	378
<i>OS Snapshot Summary Panel</i>	379
<i>Analysis Options Dialog Box</i>	379
<i>ISSnapshot.ini File</i>	379
Publication Wizard	380
<i>Welcome Panel</i>	380
<i>Publication Details Panel</i>	381
<i>Publication Data Options Panel</i>	381
<i>Publication Access List Panel</i>	382
<i>Publication Packages Panel</i>	382
<i>Publication Summary Panel</i>	383
Subscription Wizard	383
<i>Welcome Panel</i>	384
<i>Subscription Details Panel</i>	384
<i>Subscription Selection Panel</i>	385
<i>Subscription Summary Panel</i>	385
Upgrade Wizard	385
AdminStudio Database Schema Tables	386
Database Server Permissions	389
AdminStudio MSI Table Mapping	391
<i>System Tables</i>	391
<i>ConflictSolver Tables</i>	391
<i>MSI Tables</i>	392
<i>MSM Tables</i>	396
<i>OS Snapshot Tables</i>	401
<i>Workflow Tables</i>	401
<i>Security Tables</i>	402
<i>Workflow Manager Tables</i>	402
Application Catalog Replication Tables	404
<i>Tables Excluded From Replication</i>	404
<i>Tables Included in Replication</i>	405
cssys: AdminStudio System Tables	405
<i>cssysExtendedAttributeSchema</i>	405
<i>cssysSchemaInfo</i>	407
<i>cssysTables</i>	407
cstbl: AdminStudio Supporting Tables	408
<i>cstblSubscribedDatabases</i>	409
<i>cstblSubscribedData</i>	409
<i>cstblConflictConfiguration</i>	410
<i>cstblConflictACE</i>	411
<i>cstblConflictPackages</i>	412
<i>cstblConflictResults</i>	413
<i>cstblConflictTransformList</i>	414

<i>cstblConflictResultsDestPkgs</i>	414
<i>cstblPackage</i>	415
<i>cstblTransformFile</i>	416
<i>cstblGroups</i>	417
<i>cstblPackageRelationships</i>	417
<i>cstblIgnoreTableList</i>	418
<i>cstblPackageHistoryLog</i>	418
<i>cstblExtendedAttributeData</i>	419
<i>cstblGroupPackages</i>	419
as: Job Manager Tables	420
<i>AsJob</i>	420
<i>AsJobQueue</i>	421
<i>AsJobStep</i>	422
<i>AsJobStepMaster</i>	423
<i>AsJobQueueStepDetail</i>	423
<i>AsNotification</i>	424
<i>AsQueueStatus</i>	424
<i>AsSchedule</i>	425
cspch: OS Security Patch Wizard Tables	425
pdt: Predeployment Test Tables	426
Miscellaneous Tables	426
Application Manager Command-Line Functionality	426
Specifying User Interface Mode via Command Line	429
Using a Configuration File	429
<i>Application Manager Configuration File</i>	429
<i>Using a Configuration File with Command-Line Options</i>	434
Importing	434
<i>Applying Transforms and Patches During Command-Line Import</i>	435
<i>Importing Multiple Windows Installer Packages Simultaneously</i>	435
<i>Importing Multiple Merge Modules Simultaneously</i>	436
<i>Simultaneously Importing Windows Installer Packages and Merge Modules</i>	436
<i>Using the Command Line to Import All Packages in a Directory</i>	436
<i>Running Import Silently</i>	437
<i>Creating a Log File During Command-Line Import</i>	437
Connecting to Standalone Application Catalogs	437
<i>Connecting to a Specific Standalone Application Catalog Using Command-Line Options</i>	437
<i>Creating Shortcuts to Specific Standalone Application Catalogs</i>	438
Application Catalog Replication Command Line Functionality	439

Part 3: Repackaging and Customizing Installations. 441

7 Repackaging Legacy Installations Using the Repackaging Wizard 443

About Repackaging 444

Purpose of Repackaging Applications	444
Supported Legacy Installation Types	445
Repackaging 64-Bit Applications	446
Repackaging Options Comparison	447
Repackaging Wizard Best Practices	448
About Repackaging on Clean Systems	449
Alternate-Language Repackaging on Clean Machines	450
Including the InstallScript Engine With a Windows Installer Package	450
Repackaging Methods	450
Installation Monitoring Method	451
Snapshot Method	451
InstallShield Professional Logging Method	452
Configuring Repackager to Ensure Optimal Installation Capture	453
Launching Repackager Remotely	454
Installing Repackager on a Clean Machine	457
Repackaging Legacy Installations Using the Repackaging Wizard	458
Repackaging Using the Installation Monitoring Method	459
Step 1: Selecting the Repackaging Method	459
Step 2: Excluding Processes (Optional)	461
Step 3: Collecting Product Information	462
Step 4: Adding Additional Setup Programs (Optional)	463
Step 5: Set Target Project Information	464
Step 6: Set Capture Settings (Optional)	465
Step 7: Beginning the Repackaging Process	467
Repackaging Using the Snapshot Method	469
Performing Multiple Step Snapshot Repackaging	470
Step 1: Selecting the Repackaging Method	470
Step 2: Initial Analysis	472
Step 3: Install Setup and Make Manual System Changes	473
Step 4: Entering Product Information	473
Step 5: Set Target Project Information	474
Step 6: Set Capture Settings (Optional)	475
Step 7: Beginning the Repackaging Process	477
Performing Single Step Snapshot Repackaging	479
Step 1: Selecting the Repackaging Method	479
Step 2: Collecting Product Information	481
Step 3: Set Target Project Information	483
Step 4: Set Capture Settings (Optional)	483
Step 5: Beginning the Repackaging Process	485
Repackaging Using the InstallShield Professional Logging Method	488
Repackaging an InstallScript MSI Setup to a Basic MSI Setup	492
Running the Repackaging Wizard from the Command Line	493
Repackaging a Windows Installer (.msi) Package	494
Repackaging Wizard Reference	499

Repackaging Wizard.	500
Welcome Panel.	500
Method Selection Panel	501
Snapshot Method Panel	503
Collect Product Information Panel	504
InstallScript MSI Identified Panel	506
InstallShield Professional Setup Panel	508
Set Target Project Information and Capture Settings Panel	508
InstallScript MSI Conversion Output Panel	509
Repackaging Panel.	510
Summary Panel.	512
Additional Repackaging Wizard Dialog Boxes.	513
Additional Setup Programs Dialog Box	513
Setup Information Dialog Box.	514
Excluded Processes Dialog Box.	514
Analysis Options Dialog Box	515
Repackaging Wizard Command-Line Options.	516
Reboot Handling in the Repackaging Wizard	518

8 Converting Legacy Installations Using the Repackager Interface 521

About the Repackager Interface.	523
Launching the Repackager Interface	523
Setting Repackager Options	523
Selecting Data Display Colors	524
Specifying Additional Merge Module Directories	524
Controlling the Display of ICE Validation Warnings	525
Creating Repackager Projects	525
Converting Legacy Installations Using the Repackager Interface.	526
Converting Repackager 3.x Output Files	527
Converting a Microsoft SMS Project to a Repackager Project	527
Converting Novell ZENworks Projects	527
<i>Converting a Novell ZENworks Project Using the Repackager Interface.</i>	<i>528</i>
<i>Converting Multiple Novell ZENworks Projects Using the Command Line</i>	<i>529</i>
Converting WinINSTALL Projects.	531
Converting Wise Installation Projects.	531
Converting InstallShield Professional Log Files	531
Working With Repackager Projects	532
Building an InstallShield Editor Project	532
Building a Windows Installer Package	535
About the Context.msi File	540
Configuring Advanced Conversion Options.	541
Automatically Generating a Virtual Application During Repackager Project Build	543
Viewing Repackager Project Properties.	546
Using the Setup Intent Wizard to Detect File Dependencies in a Repackager Project.	547

Creating a Setup Capture Report for a Project	548
Saving Repackager Projects	549
Opening InstallShield Editor from Repackager	550
Isolating Windows Installer Packages	550
About Application Isolation.	551
Isolating Windows Installer Packages Using Application Isolation Wizard	551
About Assemblies.	552
About Manifests	552
About Digital Certificates	553
Setting Isolation Options	554
Specifying Manifest Options.	555
<i>Selecting the Assembly Type</i>	<i>555</i>
<i>Specifying the Assembly Naming Conventions</i>	<i>556</i>
Setting Digital Signature Options for Shared Assemblies	556
Building an Isolated Windows Installer Package	557
Configuring Exclusions	557
Configuring Exclusions Using Repackager	558
Excluding Files	558
Excluding All Files in a Directory	559
Excluding Directories and Subdirectories	559
Excluding Registry Keys	559
Excluding Registry Values	560
Excluding .ini Files	560
Excluding .ini File Sections	560
Excluding Shortcuts	561
Excluding All Shortcuts in a Directory	561
Excluding Shortcuts from Subdirectories	561
Specifying the External Configuration File	562
Modifying External Configuration Files	563
Configuring Exclusions Using the Exclusions Editor	563
Exclusions and Repackager	564
Exclusions and the OS Snapshot Wizard	564
Launching Exclusions Editor	564
Excluding Files	566
Excluding Files with Specific Extensions	566
Excluding Directories	567
Editing Existing File Exclusions	567
Removing File Exclusions	568
Excluding .ini Files	568
Excluding Sections from .ini Files	569
Editing Existing .ini File Exclusions	570
Removing .ini File Exclusions	570
Excluding Registry Data	571
Editing Existing Registry Exclusions	571

Removing Registry Exclusions572
Repackaging and Anti-Virus Software572
Scanning InstallShield Professional Setups for Additional Data573
Creating an InstallShield Editor Template to Use Within Repackager575
Repackager Interface Reference580
Repackager Interface581
Repackager Start Page582
Menus and Toolbar584
Dialog Boxes586
About Repackager Dialog Box587
Create Report Dialog Box587
Isolation Options Dialog Box588
Options Dialog Box591
Password Required Dialog Box592
Project Properties Dialog Box593
WinINSTALL Conversion Dialog Box595
Repackager Views596
Captured Installation View596
Files and Folders View599
Registry Entries View600
Shortcuts View602
INI Files View603
Deleted Files View604
Deleted Registry Entries View605
Repackaged Output View606
Package Information View611
Advanced Settings View612
SmartScan Wizard614
Welcome to the SmartScan Wizard Panel615
Original InstallShield Professional Setup Panel616
Scanning Panel617
Setup Feature Tree Panel617
Scanning Media Panel618
Setup Intent Wizard619
Welcome Panel619
Scanning Project Panel620
Results Panel620
VMware Repackaging Wizard621
Welcome Panel621
VMware Virtual Machines Panel621
Exclusions Editor Interface622
Menus622
Files Tab623
.ini Files Tab624

Registry Tab	625
File Exclusion Information Dialog Box	626
INI File Exclusion Information Dialog Box	627
Choose Registry Key Dialog Box	627
Edit Registry Key Dialog Box	628
About Exclusions Editor Dialog Box	628
Options.ini File	628
Files Associated with Repackager	635
Repack.ini File	638
Using InstallShield to Chain Multiple Windows Installer Packages Together	639
Troubleshooting	639
Troubleshooting Guidelines for WinINSTALL Conversion	639
Troubleshooting Guidelines for SMS Conversion	640
Resolving an "Error Building Table File" Error	640
9 Converting Windows Installer Packages to Virtual Applications	643
Getting Started With Application Virtualization	644
About Application Virtualization	646
Types of Virtual Package Conversion	648
About Microsoft Application Virtualization	651
About Microsoft Application Virtualization (App-V)	651
Creating 64-Bit App-V Packages	655
Components of an App-V Application	656
How Windows Services Are Integrated into an App-V Application	658
About ThinApp Virtual Packages	659
About ThinApp Applications	659
ThinApp Virtual Operating System	660
Components of a ThinApp Application	660
Benefits of Deploying ThinApp Applications	662
Prerequisites for Building a ThinApp Application	662
About Citrix Virtual Packages	662
About Citrix XenApp and Citrix Profiles	663
About Citrix XenApp	663
About Citrix Profiles (.profile)	664
Benefits of Deploying Citrix Profiles	666
Converting a Single Application to a Virtual Package	668
Converting a Suite of Applications into a Virtual Package	670
Converting a Directory of Packages into Individual Virtual Packages	672
Testing Windows Installer Packages for Virtualization Compatibility	673
Using Windows Installer Converter to Test Packages for Virtualization Compatibility	674
About Rules and Tests	677
Pre-Conversion Evaluation Rules Shipped With AdminStudio	677
Extensible Test Framework: Writing Your Own Rules for Virtualization Conversion	678

About Custom Rules678
Components of a Rule679
Creating Your First Custom Rule680
Custom Rules Reference681
Rule Data Binding690
Testing App-V Applications Using the App-V Application Launcher691
Running the Windows Installer Converter from the Command Line693
Capturing Virtualization Context.694
Windows Installer Converter Reference695
Windows Installer Converter.695
Welcome Panel696
Select Conversion Target Panel697
Conversion Mode Panel698
Select Windows Installer Package700
Select a Directory Panel701
Select a Suite of Packages Panel.702
Test Package(s) Panel704
Edit Windows Installer Package Panel705
Package Conversion Panel706
Virtualization Conversion Error Messages708
Error -9000: Unknown Exception708
Error -9001: Unknown COM708
Error -9002: Error Opening Package709
Error -9003: Error Saving Package.709
Error -9004: Process Cancelled By User710
Error -9005: Error Creating Temporary Folder710
Error -9006: Error Decompressing Package710
Error -9007: File With Extension Not Found.711
Error -9008: Error Extracting Icon.711
Error -9009: Unknown Provider712
Error -9010: Invalid Target File Name.712
Error -9011: Error Reading MSI Table713
Error -9012: Unexpected Error in Method713
Error -9013: Type Library Not Found713
Error -9014: ShellExecute Failed714
Error -9015: Unable to Determine Full Path for Driver714
Warning -9016: Contents of Table Ignored715
Warning -9017: .NET 1.x Assembly Not Supported.716
Warning -9018: Custom Actions Ignored716
Warning -9019: Conditionalized Components717
Error -9020: Directory With Null Parent718
Error -9021: Unable to Extract COM Data718
Error -9022: Complus Table719
Error -9024: FileSFPCatalog.719

Warning -9026: LaunchCondition Table	720
Warning -9027: LockPermissions Table	720
Error -9028: MoveFile Table	721
Error -9029: MsiDriverPackages Table	722
Warning -9030: ODBCTranslator Table	722
Warning -9031: RemoveFile Table	723
Warning -9032: RemoveIniFile Table	723
Warning -9033: RemoveRegistry Table	724
Error -9036: ISCEInstall Table	724
Error -9037: ISComPlusApplication Table	725
Error -9038: ISPalmApp Table	725
Error -9039: ISSQLScriptFile Table	726
Error -9040: ISVRoot Table	726
Error -9041: ISXmlFile Table	727
Error -9051: Package Decompression Canceled	727
Error -9100: CreateInstance of Package Object Failed	727
Error -9101: Create Operation of Package Object Failed	728
Error -9102: Failed to Write Header Information	728
Error -9103: Citrix Finalization Failed	729
Error -9104: Citrix Save Failed	729
Error -9105: Error Initializing Citrix Writer	729
Error -9106: Error Initializing Citrix Package	730
Error -9107: Error Writing Citrix File Entries	730
Error -9108: Error Determining Source File Path	731
Error -9109: Error Writing Citrix Folder Entries	731
Error -9110: Error Writing Citrix Registry Entries	731
Error -9113: Error Writing Citrix INI File Entries	732
Error -9114: Error Writing Citrix Shortcuts	732
Error -9115: Error Saving Citrix Profile	733
Error -9116: Error Creating Empty Citrix Profile	733
Error -9117: Error Creating Intermediate Folder	733
Error -9118: Error Initializing Citrix Profile	734
Error -9119: Error Creating Default Target in Citrix Profile	734
Error -9120: Error Deleting File From Profile	734
Error -9121: Failed to Copy File into Citrix Profile	735
Error -9122: Target Does Not Exist in Citrix Profile	735
Error -9124: No Shortcuts Created for this Profile	736
Error -9125: Error Writing Citrix File Type Associations	736
Error -9126: Failed to Sign Profile Using Certificate	737
Error -9127: Could Not Create Script Execution	737
Warning -9128: Duplicate Shortcut	737
Warning -9129: Duplicate Shortcut Names	738
Warning -9130: Duplicate Shortcut Targets	738
Warning -9131: Unable to Resolve Installer Variable	739

Warning -9132: 16 Color Shortcut Icon Not Found	739
Warning -9133: Shortcut Icon Not Found	739
Warning -9134: Failure to Extract Icon from Executable	740
Error -9135: Shortcut Target is 16-Bit.	740
Warning -9136: Some Files May Not Be Decompressed	741
Warning -9137: Destination Directory Cannot Be Found.	741
Warning -9138: Ignoring a DuplicateFile Table Entry	742
Error -9200: ThinApp Must Be Installed	742
Warning -9201: Extension for Shortcut Files Must Be ".exe"	743
Error -9202: No Applications Were Created.	743
Error -9203: ThinApp Tool is Missing	744
Error -9204: Duplicate Shortcut	744
Error -9205: Identically-Named Shortcut Already Exists, But With Different Parameters	744
Error -9206: Identically-Named Shortcut Already Exists, But With a Different Target.	745
Error -9207: Error During Build Process (vregtool.exe)	745
Error -9208: Error Occurred During Build Process (vftool.exe)	746
Error -9209: Error Occurred During Build Process (tlink.exe)	746
Error -9300: Unhandled Exception During AdviseFile Operation	746
Error -9301: Unhandled Exception During AdviseRegistry Operation.	747
Error -9302: Unhandled Exception During Command Action	747
Error -9303: Unhandled Exception During Alter File Action.	748
Error -9304: Unhandled Exception During Alter Registry Action.	748
Error -9305: Unhandled Exception During Create Action	748
Error -9306: Unhandled Exception During Execution of Rules Engine	749
Error -9401: Error Initializing App-V Writer	749
Error -9402: Error Initializing App-V Package.	749
Error -9403: Error Writing App-V File Entries	750
Error -9404: Error Writing App-V Folder Entries.	750
Error -9405: Error Writing App-V Registry Entries	750
Error -9406: Error Writing App-V INI File Entries	751
Error -9407: Error Writing App-V Shortcuts	751
Error -9408: Error Writing App-V File Type Data	751
Error -9409: Error Saving App-V Data	752
Error -9410: Error Determining Source File Path	752
Error -9411: OSD File Template Could Not Be Extracted	752
Error -9412: OSD File Could Not Be Saved	753
Error -9413: App-V OSD Save	753
Warning -9414: Local App-V Application Specified as a Dependency of the Primary Application.	753
Error -9415: Dependency Application Was Not Found	754
Warning -9416: Invalid Primary Application Directory	754
Error -9417: Dependency Application's OSD File Contains an Invalid HREF Value	755
Error -9418: Error While Privatizing Side-By-Side Assemblies	755
Error -9419: Error Inserting Watermark	755
Warning -9500: Shortcut Missing	756

Error -10000: Process Cancelled By User	756
Warning -10001: Suite File Missing	757
Warning -10002: Suite File is Duplicate	757
Warning -10003: Application File Missing	757
Warning -10004: INI File Missing	758
Fix 11000: Excluding TCPIP Registry Entries	758
Fatal Error 11001: Fail on VMware	759
Warning 11003: Control Panel Applet - Citrix	759
Fix 11004: Control Panel Applet - ThinApp	759
Fatal Error 11005: QuickTime 7.4.1 Causes Fatal Error	760
Fix 11006: Adobe Distiller Exclude AdobePDFSettings	760
Fix 11007: Exclude URL Shortcut	760
Steps to Take Before Calling Technical Support	761
Application Features Requiring Pre- or Post-Conversion Actions	761

10 Creating Customized Virtual Applications 765

About Virtualization766

About the AdminStudio Virtualization Interface768

About the Virtualization Assistant Tabs	768
Using the More Options, Other Places, and Help Links Sections in a Virtualization Assistant	769
Navigating in a Virtualization Assistant	770
Opening the Installation Designer	770
Showing or Hiding the Virtualization Assistants	771

Creating Microsoft App-V Applications771

Overview of Microsoft Application Virtualization and the Microsoft App-V Assistant	772
About Microsoft Application Virtualization (App-V) and the App-V Assistant	773
Components of an App-V Application	776
About the Microsoft App-V Assistant	778
<i>Process for Authoring an App-V Application Using the Microsoft App-V Assistant</i>	779
<i>Supported InstallShield Project Types</i>	781
<i>How Transforms are Included in an App-V Application</i>	781
<i>How Windows Services Are Integrated into an App-V Application</i>	782
<i>Quick Start for Microsoft App-V Sequencer Users</i>	783
Using the Microsoft App-V Assistant to Create an App-V Application	786
Specifying Package Information and Deployment Options	786
<i>Specifying Package Information</i>	787
<i>Specifying Operating System Requirements</i>	788
<i>Specifying Upgrade Package Information</i>	788
<i>Specifying the Deployment Server</i>	789
<i>Including Diagnostic Tools With an App-V Application</i>	790
Managing Files in an App-V Application	792
<i>Adding, Deleting, and Moving Files and Folders in an App-V Application</i>	792
<i>Controlling the Display of Predefined Folders</i>	796
<i>Specifying the Primary Application Directory</i>	798

Setting Isolation Options for Folders and Files	800
Overview of App-V Isolation Options	800
Setting Isolation Options for Folders and Files	801
Inheritance of Isolation Options from Folders to Files	802
Modifying Shortcuts to the App-V Application's Executable Files	803
App-V Applications and the Virtual Environment	804
App-V Shortcut Requirements	804
Creating a New App-V Application	804
Including an Existing App-V Application	805
Excluding or Deleting an Existing App-V Application	805
Excluding vs. Deleting App-V Application Shortcuts	807
Renaming an App-V Application	807
Modifying App-V Application Registry Settings	807
About the Windows Registry	807
Adding or Deleting Registry Keys and Values	808
Setting App-V Application Registry Isolation Options	808
Inheritance of Isolation Options in the Registry	809
Performing Dynamic Suite Composition	810
Modifying Build Options	811
Selecting Releases to Build	812
Enabling App-V Application Building When in Direct Edit Mode	812
Specifying Whether to Compress the Data Files in an App-V Package	813
Including Additional Windows Installer Packages in an App-V Application	813
Building a Windows Installer Package to Assist in the Distribution of an App-V Application	814
Specifying Package Feature Block Optimizations	814
Building an App-V Application	816
Testing an App-V Application Using the App-V Application Launcher	818
Microsoft App-V Assistant Reference	819
Pages	820
Microsoft App-V Assistant Home Page	820
Package Information Page	821
Files Page	826
Applications Page	829
Registry Page	831
Dynamic Suite Composition Page	832
Build Options Page	835
Dialog Boxes	839
Advanced Settings Dialog Box	839
App-V Diagnostic Tools Dialog Box	841
App-V Package Upgrade Settings Dialog Box	842
File Isolation Options Dialog Box	843
Folder Isolation Options Dialog Box	844
Launch App-V Application Dialog Box	845
Package Optimizations Dialog Box	845

<i>Primary Application Directory Dialog Box</i>	847
<i>Registry Isolation Options Dialog Box</i>	848
Building App-V Applications Using the Command Line	849
App-V Application Conversion Error and Warning Messages	850
Application Features Requiring Pre- or Post-Conversion Actions	850
Creating ThinApp Applications	850
Overview of the ThinApp Assistant	851
About ThinApp Applications	851
<i>The ThinApp Virtual Operating System</i>	851
<i>Benefits of Deploying ThinApp Applications</i>	852
About the ThinApp Assistant	852
<i>Process for Authoring a ThinApp Application Using the ThinApp Assistant</i>	853
<i>Components of a ThinApp Application</i>	855
<i>Supported InstallShield Project Types</i>	857
<i>How Transforms are Included in a ThinApp Application</i>	857
<i>About Sandboxes</i>	858
Using the ThinApp Assistant to Create a ThinApp Application	858
Specifying ThinApp General Settings	859
<i>Specifying Sandbox Information</i>	859
<i>Specifying Control Access via Active Directory</i>	860
<i>Prerequisites for Building a ThinApp Application</i>	861
<i>Including Diagnostic Tools With a ThinApp Application</i>	861
Managing Files and Folders in a ThinApp Application	863
<i>Adding, Deleting, and Moving Files and Folders in a ThinApp Application</i>	863
<i>Controlling the Display of Predefined Folders</i>	867
Setting ThinApp Isolation Options	869
<i>Overview of ThinApp Isolation Options</i>	869
<i>Setting Isolation Options for Folders</i>	871
<i>Inheritance of Isolation Options from Folders to Files</i>	872
Modifying Shortcuts to the ThinApp Application's Executable Files	872
<i>ThinApp Applications and the Virtual Environment</i>	874
<i>Compressing a ThinApp Application</i>	874
<i>ThinApp Shortcut Requirements</i>	875
<i>Creating a New ThinApp Application</i>	875
<i>Including an Existing ThinApp Application</i>	876
<i>Excluding or Deleting an Existing ThinApp Application</i>	876
<i>Excluding vs. Deleting ThinApp Application Shortcuts</i>	878
<i>Renaming a ThinApp Application</i>	878
Modifying ThinApp Application Registry Settings	878
<i>About the Windows Registry</i>	878
<i>Adding or Deleting Registry Keys and Values</i>	879
<i>Setting ThinApp Isolation Options on Registry Keys</i>	879
<i>Inheritance of ThinApp Isolation Options in the Registry</i>	880
Modifying Build Options	881

<i>Selecting Releases to Build</i>	881
<i>Enabling ThinApp Application Building When in Direct Edit Mode</i>	882
<i>Including Additional Windows Installer Packages in a ThinApp Application</i>	882
<i>Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application</i>	883
<i>Setting ThinApp Log Monitor Tracing Options</i>	883
<i>Setting AppLink Options</i>	885
<i>Setting AppSync Options</i>	888
Building a ThinApp Application	891
ThinApp Assistant Reference	893
Pages	893
ThinApp Assistant Home Page	894
General Settings Page	896
Files & Folders Page	898
Applications Page	901
Registry Page	902
Build Options Page	904
Dialog Boxes	910
ThinApp Diagnostic Tools Dialog Box	910
Folder Isolation Options Dialog Box	912
Registry Isolation Options Dialog Box	913
AppLink Settings Dialog Box	915
Add AppLink Reference Dialog Box	918
AppSync Settings Dialog Box	921
Building ThinApp Applications Using the Command Line	925
ThinApp Application Conversion Error and Warning Messages	925
Application Features Requiring Pre- or Post-Conversion Actions	925
ThinApp Not Found	925
ThinApp Application Configuration File: package.ini	925
[BuildOptions]	926
[Compression]	931
[Isolation]	931
[MainApp.exe]	932
[Test.exe]	935
Creating Citrix Profiles	935
Overview of the Citrix Assistant	935
About Citrix XenApp	937
About the Citrix Assistant	938
About Citrix Profiles	939
Benefits of Deploying Citrix Profiles	940
Supported InstallShield Project Types	942
How Transforms are Included in a Citrix Profile	942
Using the Citrix Assistant to Create a Citrix Profile	943
Specifying Citrix Profile Information	943
Specifying the Profile Name, Description, and Version	944

Enabling a Citrix Profile to Run Non-Included Executables	945
Including Diagnostic Tools With a Citrix Profile	945
Specifying Operating System and Language Requirements	947
Setting Operating System Requirements and Service Pack Levels	947
Setting Language Requirements	948
How Requirements are Applied at Runtime	949
Adding Pre-Launch and Post-Exit Scripts	950
Managing Files and Folders in a Citrix Profile	952
Managing Files and Folders in a Citrix Profile	952
Controlling the Display of Predefined Folders	956
Setting Isolation Options	957
Overview of Citrix Isolation Options	958
Setting Isolation Options for Folders and Files	960
Inheritance of Isolation Options from Folders to Files	960
Modifying Profile Shortcut Settings	961
Shortcuts and the Isolation Environment	962
Shortcut Requirements	963
Creating a New Profile Shortcut	964
Including an Existing Profile Shortcut	964
Excluding vs. Deleting a Profile Shortcut	965
Conditions When a Shortcut Should be Excluded or Deleted	966
Renaming a Shortcut	966
Modifying Profile Registry Settings	967
About the Windows Registry	967
Adding or Deleting Registry Keys and Values	968
Setting Registry Isolation Options	968
Inheritance of Isolation Options in the Registry	969
Modifying Build Settings	969
Selecting Releases to Build	970
Digitally Signing a Citrix Profile	971
Including Additional Windows Installer Packages in a Citrix Profile	971
Enabling Citrix Profile Building When in Direct Edit Mode	972
Building a Citrix Profile	972
Citrix Assistant Reference	974
Pages	974
Home Page	975
Profile Information Page	976
Profile Requirements Page	978
Profile Files Page	981
Profile Shortcuts Page	984
Profile Registry Page	986
Build Settings Page	987
Dialog Boxes	990
Script Execution Dialog Box	991

<i>Diagnostic Tools Dialog Box</i>	991
<i>File Isolation Options Dialog Box</i>	993
<i>Folder Isolation Options Dialog Box</i>	994
<i>Registry Isolation Options Dialog Box</i>	996
<i>Service Packs Requirement Dialog Box</i>	998
Building Citrix Profiles Using the Command Line	999
Citrix Profile Conversion Error and Warning Messages	999
Application Features Requiring Pre- or Post-Conversion Actions	999

11 Customizing and Authoring Installations Using InstallShield 1001

AdminStudio-Specific Functionality in InstallShield Editor	1002
InstallShield Editor Integration with Application Manager and the Software Repository	1004
Getting a Copy of a Software Repository Package	1005
Opening and Editing a Software Repository Package	1006
Adding a Package to the Software Repository via the InstallShield Editor Build Process	1008
Using the Application Manager Context Menu	1008
Application Manager Tab on InstallShield Editor Options Dialog Box	1009
Microsoft App-V, VMware ThinApp, and Citrix XenApp Virtualization Support	1009
Differences Between InstallShield Editor and InstallShield 2009 Professional Edition	1010
InstallShield Editor Help Library	1010

12 Customizing Installations Using Tuner. 1011

Working with Transforms	1012
Creating New Transform Files	1013
Opening Existing Transforms	1014
Opening Recently Accessed Transforms	1015
Creating Generic Transforms	1015
Using Response Transforms	1016
Viewing Transform Properties	1016
Validation	1016
Prevalidating Windows Installer Packages	1017
Handling Invalid Windows Installer Packages	1018
Postvalidating Transforms	1019
Evaluation Files and Internal Consistency Evaluators	1020
Setup Organization	1020
Changing a Feature's Visibility	1021
Setting the Initial State of a Feature	1021
Editing a Feature's Description	1022
Setting the Default Destination	1022
Setting the Default Organization	1022
Changing the Destination Variable	1023
Preventing Features from Displaying During Custom Installation	1023
Setting Feature Properties	1023

Using Feature Advertisement	1024
Configuring Package Content.....	1025
Files and Folders	1025
Adding Files	1025
Displaying Files from the Base Windows Installer Package	1026
Preventing Installation of Files from the MSI	1026
Removing Added Files	1027
Storing Added Files	1027
Registry Entries.....	1027
Creating a Registry Key	1028
Creating a Registry Value	1029
Importing REG Files	1029
Removing Registry Information	1030
Shortcuts.....	1030
Creating Shortcuts	1031
Changing a Shortcut's Icon	1031
Change a Shortcut's Location	1032
Changing a Shortcut's Target.....	1032
Creating a Hot Key	1032
Removing Shortcuts	1033
Determining the Path of Changed Shortcuts	1033
INI Files	1033
Adding INI Files.....	1034
Importing Existing INI Files	1034
Adding Sections to INI Files	1035
Adding New Keys to INI File Sections	1035
Modifying INI File Keys, Values, and Actions.....	1035
Removing INI Files	1036
Removing Sections from INI Files	1037
Removing INI File Section Keys	1037
ODBC Resources	1037
Adding New Data Sources	1038
Adding New ODBC Data Source Attributes	1038
Adding New ODBC Driver Attributes	1038
Editing ODBC Data Source Attributes	1039
Editing ODBC Driver Attributes.....	1039
Removing Existing ODBC Data Sources	1039
Removing ODBC Driver Attributes	1039
Removing ODBC Data Source Attributes.....	1040
NT Services.....	1040
Setting NT Service Arguments	1040
Setting NT Service Dependencies	1041
Setting the NT Service Description.....	1041
Setting the NT Service Display Name	1041

Setting the NT Service Error Control Level	1042
Setting the NT Service Load Order Group	1042
Setting the NT Service Overall Install Result	1042
Setting the NT Service Start Type	1043
Setting NT Service Start Name and Password	1043
Setting the NT Service Type	1043
Working with Dialogs	1044
Hiding Dialogs During UI Sequences	1044
Restoring Dialog Sequences	1045
Suppressing the License Agreement Dialog Box	1045
Disabling Custom Setups	1045
Editing Dialog Properties	1046
Dialogs View vs. Command-Line Options	1047
Dialog Suppression Issues	1047
Configuring Additional Server Locations	1047
Adding Additional Server Locations	1048
Modifying Server Locations	1048
Removing Server Locations	1048
Reordering Server Locations	1049
Changing Add/Remove Program Settings	1049
Changing Add/Remove Programs Properties	1049
Disabling the Modify, Remove, or Repair Buttons	1050
Customizing Setup Properties	1050
Adding Custom Setup Properties	1050
Adding and Editing Comments	1051
Removing Custom Setup Properties	1051
Modifying Setup Properties	1051
Preparing Packages for Distribution	1052
Copying the Installation to a Network Location	1052
Copying the Installation to an FTP Server	1053
Creating a Package Definition File (PDF)	1053
Creating an SMS File	1054
Instructing SMS to Create a Management Information Format File at Deployment Time	1055
Deploying Windows Installer Setup Packages with Systems Management Server 2.0	1055
Creating a Setup.exe File for the Package and Transform	1055
Additional Setup.ini Parameters	1056
Directly Editing Packages	1056
Adding a New Record Using the Direct Editor	1057
Finding and Replacing Using the Direct Editor	1057
Launching the Direct Editor from the Validation Tab	1057
Tuner Reference	1058
User Interface Reference	1058
Menus and Toolbar	1059

View Bar	1062
Checklist	1062
<i>Customization Steps Checklist</i>	1062
Output Window	1063
Customize Dialog Box	1064
Properties Dialog Box	1064
Options Dialog Box	1064
Transform Summary Dialog Box	1064
Dialog Properties Dialog Box	1066
Tuner Views	1067
Tuner Start Page	1068
Welcome to Tuner View	1068
Create a New Transform View	1068
Open a Recent Transform View	1070
Open an Existing Transform View	1071
Help View	1071
Package Validation View	1071
Prevalidation View	1071
Organization View	1072
Product Properties View	1072
Features View	1073
System Configuration View	1074
Files and Folders View	1075
Registry View	1075
Shortcuts View	1076
Shortcuts View/Shortcut Properties	1077
Shortcuts View/Shortcut Target	1079
Shortcuts View/Shortcut Locations	1080
INI Files View	1081
ODBC Resources View	1082
NT Services View	1083
Application Configuration View	1085
Server Locations View	1085
Setup Properties View	1086
Dialogs View	1086
Add/Remove Programs View	1086
Package Preparation View	1087
Postvalidation View	1088
Package View	1089
Package View/Location View	1090
Package View/Setup View	1090
Package View/SMS View	1090
Additional Tools View	1090
Direct Editor	1090

Import INI File Wizard	1092
Welcome Panel	1092
Import INI File Panel	1092
Import Conflict Options Panel	1092
Finishing INI File Import Panel	1093
Import REG File Wizard	1093
Welcome Panel	1093
Import Registry File Panel	1093
Import Conflict Options Panel	1093
Finishing Registry Import Panel	1094
Packaging Wizard	1094
Location Panel	1094
Setup.exe Panel	1095
SMS Panel	1095
Packaging Summary Panel	1095

Part 4: Testing Windows Installer Packages..... 1097

13 Identifying and Resolving Application Conflicts Using ConflictSolver 1099

About ConflictSolver	1100
Application Manager and ConflictSolver	1100
Functionality Comparison	1101
Accessing the Tools	1102
Displaying ConflictSolver Options in Application Manager	1102
Creating and Connecting to Application Catalogs	1102
Types of ACE Rules: Best Practice and Conflict	1103
Connecting to Existing Application Catalogs	1103
Best Practices for Conflict Detection	1104
Plan Your Import	1104
Plan Your Conflict Detection	1106
Perform Conflict Detection	1106
Scanning for Dependencies	1108
Validating Packages	1109
Performing Validation	1110
Validating Before Import	1110
Validating During Import	1111
Validating After Import	1112
Validating an Imported Package	1112
Viewing ICE Error Information	1113
Suppressing an ICE Error	1114
Deleting an ICE Error	1115
Changing the Default Validation File	1116
Excluding Specific ICEs from Execution During Validation	1116

Handling Invalid Windows Installer Packages	1117
About ICE43, ICE50, and ICE57 Validation Rules for Shortcuts	1117
Identifying and Resolving Conflicts	1118
Using Multiple Source Packages in an Analysis	1119
Setting Options	1119
Changing Default Conflict Types Checked	1119
Changing Resolution Options	1121
Checking for Conflicts Across All Source and Target Packages	1121
Checking for Conflicts	1122
Checking for Conflicts Using the Conflict Wizard	1122
Checking for Conflicts During Import	1124
Viewing Identified Conflicts in the Conflicts View	1125
Running Conflict Identification Again	1126
Deleting Persisted Conflict Data	1126
Conflict Persistence	1126
Resolving Conflicts	1127
Conflict Resolution Process	1127
Conflict Resolution and the Software Repository	1128
Automatically Resolving Conflicts	1129
Manually Resolving Conflicts	1130
Resolving Conflicts Directly in Windows Installer Packages	1131
Using Transforms for Conflict Resolution	1131
Reimporting Packages after Successful Conflict Resolution	1132
Generating Reports	1132
Generating Package Reports	1134
Generating File and Registry Reports	1134
Generating Reports in Crystal Reports Format	1135
Creating Custom Reports	1136
Step 1: Creating an ODBC File DSN	1136
Step 2: Creating the Base Report	1137
Step 3: Configuring Basic Settings	1137
Step 4: Creating Page Column Headings	1138
Step 5: Adding Database Fields	1138
Step 6: Adding Parameters	1139
Step 7: Creating Conditions for Filtering	1140
Step 8: Modifying the Footer	1140
Step 9: Using the Report in ConflictSolver	1140
Application Conflict Evaluators (ACEs)	1141
ACE Index	1142
ACE02	1146
ACE03	1146
ACE04	1147
ACE05	1148
ACE06	1149

Contents

ACE07	1149
ACE08	1151
ACE09	1152
ACE10	1153
ACE12	1155
ACE13	1157
ACE14	1158
ACE15	1159
ACE16	1160
ACE17	1161
ACE18	1162
ACE19	1163
ACE20	1163
ACE21	1164
ACE22	1165
ACE23	1167
ACE24	1168
ACE25	1170
ACE26	1170
ACE27	1171
ACE28	1172
ACE29	1173
ACE30	1173
ACE31	1174
ACE32	1175
ACE33	1176
ACE34	1176
ACE35	1177
ACE36	1178
WTS01	1179
WTS02	1179
WTS03	1180
WTS04	1181
WTS05	1182
User-Defined ACEs	1182
Types of User-Defined ACEs	1183
Creating User-Defined ACEs	1184
<i>Creating a Custom/Source Only Packages ACE</i>	<i>1184</i>
<i>Creating a Custom/Source and Target Packages ACE</i>	<i>1186</i>
<i>Creating a User Provided DLL-Based ACE</i>	<i>1189</i>
Editing User-Defined ACEs	1191
Deleting User-Defined ACEs	1192
Viewing ACE Metrics	1192
Location of ACE Files	1194

Conflict Application Resolution Definitions (CARDS)	1194
CARD Index	1195
CARD02	1196
CARD04	1196
CARD05	1197
CARD06	1197
CARD07	1197
CARD15	1198
CARD18	1198
CARD19	1198
CARD20	1199
ConflictSolver Reference	1199
ConflictSolver Interface	1199
Menus and Toolbar	1200
Product View Icons	1203
Context Menus	1206
Output Window	1208
Views	1211
Product View	1211
Conflicts View	1213
Validation View	1215
Test Results View	1217
Patch Impact View (Products Tab)	1217
Dependencies View	1218
Associated Patches View	1219
Tables View	1219
OS Snapshot View	1219
Tables View for OS Snapshots	1220
Other Setup Types View	1220
Tables View for Other Setup Types	1221
NCP Views	1221
Conflicts View for Marimba NCP Files	1222
Tables View for Marimba NCP Files	1222
Patches Tab Views	1222
Patches Group View	1222
New Patches Group View	1223
Group View of a Selected Group	1224
Patch View	1225
Impact Analysis View (Patches Tab)	1226
Dialog Boxes	1227
About ConflictSolver Dialog Box	1227
ACE Rule Properties Dialog Box	1227
General Information Tab	1228
Additional Information Tab	1229

<i>Custom Options Tab</i>	1230
<i>Where Clause Tab</i>	1230
<i>DLL Information Tab</i>	1231
Add Ignore Table Dialog Box	1231
CARD Resolution Options Dialog Box	1232
Conflict Information Dialog Box	1232
Conflict Resolution Dialog Box	1232
Customize Dialog Box	1235
Properties Dialog Box	1235
Expression Builder Dialog Box	1236
Options Dialog Box	1237
<i>Rules Tab</i>	1237
<i>Duplicate Package Tab</i>	1239
<i>Extended Attributes Tab</i>	1241
<i>General Tab</i>	1241
<i>Import Tab</i>	1242
<i>Resolution Tab</i>	1243
<i>Validate Tab</i>	1244
<i>History Tab</i>	1246
Patch Impact Information Dialog Box	1246
Resolution Details Dialog Box	1247
Resolution Operations Dialog Box	1247
Resolution Options Dialog Box	1248
Rules Viewer Dialog Box	1248
Wizards	1248
Conflict Wizard	1248
<i>Welcome Panel</i>	1249
<i>Source Type Panel</i>	1250
<i>MSI Source Information Panel</i>	1250
<i>Source Package Panel</i>	1250
<i>Choose Action Panel</i>	1251
<i>Best Practice Rules Panel</i>	1251
<i>Conflict Rules Panel</i>	1252
<i>Target Information Panel</i>	1252
<i>Summary Panel</i>	1253
Rules Wizard	1253
<i>Welcome Panel</i>	1253
<i>General Information Panel</i>	1254
<i>Additional Information</i>	1254
<i>Custom Options Panel</i>	1256
<i>Token Grammar</i>	1257
<i>Where Clause Panel</i>	1257
<i>DLL-Based ACEs Panel</i>	1258
<i>Summary Panel</i>	1258

Validation Wizard	1258
MSI Source Information Panel	1259
Summary Panel	1259
14 Identifying and Resolving Package Errors Using PackageExpert	1261
About PackageExpert	1262
Configuring PackageExpert	1264
Setting Default Test Configuration	1264
Specifying the Resolution Method	1265
Testing Packages and Viewing Results	1266
Testing Packages	1266
Testing External Packages	1267
Testing Packages in an Application Catalog	1268
Testing Packages From ConflictSolver	1269
Viewing and Managing Test Results	1271
Viewing Test Results in PackageExpert	1272
Information Shown for an Error Message	1272
Test Result Views	1273
Publishing Test Results to the Application Catalog	1278
Publishing Test Results When Prompted	1279
Toggling Between Local and Published Test Results	1279
Viewing Test Results in ConflictSolver	1280
Deleting Test Results	1281
Customizing Test Results	1282
Changing the Severity of a Specific Error	1282
Suppressing the Display of Specific Errors	1287
Adding Ad-Hoc Test Results	1290
Resolving Errors	1293
Adding Manual Tests	1294
About Manual Tests	1294
Creating a Manual Test	1296
Running and Resolving a Manual Test	1297
Sample Manual Test	1299
PackageExpert Reference	1301
PackageExpert Interface	1302
PackageExpert Home Page	1302
PackageExpert Package Tree	1303
PackageExpert Toolbar Menus	1304
PackageExpert Icons	1305
Status Bar	1307
PackageExpert Views	1309
Test and Resolve View	1309
Configuration View	1312

Filter View	1313
Options View	1313
PackageExpert Dialog Boxes	1314
Connect Application Catalog Dialog Box	1314
Enterprise Server Tab	1315
Standalone Tab	1316
Custom Test Results Editor Dialog Box	1317
Manual Test Perform Dialog Box	1319
Manual Test Resolve Dialog Box	1320
Select a Package Dialog Box	1321
Test Result Filter Editor Dialog Box	1322
Unpublished Packages Dialog Box	1323

15 Analyzing the Impact of Installing Microsoft Operating System Patches 1325

About Microsoft Operating System Patches 1326

Using OS Security Patch Wizard to Import Microsoft OS Patches 1328

Identifying and Downloading Microsoft Operating System Patch Files	1329
Importing a Microsoft Operating System Security Patch Into the Application Catalog	1332
About Dual-Mode OS Patches	1335

Analyzing the Impact of Installing a Microsoft Operating System Patch 1335

Performing Patch Impact Analysis	1336
Viewing Patch Impact Analysis Results	1337
Viewing Patch and Patch Impact Information in Application Manager and ConflictSolver	1339
Generating the Patch Report	1343

Reference 1344

Patch Impact Analysis Wizard	1344
Welcome Panel	1344
OS Snapshot Panel	1345
Source Patches Panel	1345
Target Products Panel	1346
Summary Information Panel	1346
OS Security Patch Wizard	1346
Welcome Panel	1347
Patch File Selection Panel	1347
Patch File Branch Selection Panel	1348
Patch Bulletin Information Panel	1349
Additional Patch Information Panel	1349
Patch Import Summary Panel	1350
Import Process Results Panel	1350
Patch Properties Dialog Box	1350
General Tab	1351
Contents Tab	1351
Products Tab	1352

16 Isolating Applications Using Application Isolation Wizard	1353
About Application Isolation Wizard	1354
Isolating Repackaged Setups Using Repackager	1355
Launching the Application Isolation Wizard	1355
Isolation Methods	1356
Assemblies	1356
Manifests	1357
Digital Signatures	1358
Certificates	1358
Code Signing Technologies	1358
Software Publishing Credentials	1359
Certificate Store	1359
Private Keys	1359
Isolating Applications	1359
Setting Assembly Naming Conventions	1360
Modifying the Default Isolation Recommendations	1361
Filtering File Listings when Manually Configuring Isolation	1362
Servicing Published Shared Assemblies	1363
Application Isolation Wizard Reference	1363
Welcome Panel	1363
Windows Installer File Selection Panel	1364
Isolation Method Panel	1364
Summary Information Panel	1364
Application Isolation Progress Panel	1365
Completing the Application Isolation Wizard Panel	1365
Advanced Options Dialog Box	1365
Manifest Options Tab	1366
Digital Signature Tab	1367
Manifest and Assembly Design Dialog Box	1368
Isolated Components Design Dialog Box	1368
Assembly Properties Dialog Box	1368
Application Manifest Properties Dialog Box	1369
Command-Line Options	1370
Configuration Files	1370
Manifest Examples	1372
17 Ensuring Package Quality Using QualityMonitor	1373
About QualityMonitor	1374
Creating New QualityMonitor Project Files	1374
Opening Existing QualityMonitor Project Files	1375
Working with Test Cases	1375

Running Individual Test Items	1376
Running Multiple Test Items	1377
Adding Test Item Comments	1378
Adding Test Case Comments	1379
Viewing Test Item Details	1379
Clearing Test Case Results	1380
Manually Setting Test Case Status	1381
Manually Setting Test Item Status	1382
Filtering Test Case Data	1382
Deployment Testing	1383
Checking Class IDs	1384
Checking File Associations	1385
Checking Help Files	1385
Checking Prog IDs	1386
Checking Services	1386
Checking Shortcuts	1387
Checking Type Libraries	1387
Checking Manifests	1388
Checking ODBC Data Sources	1389
Checking ODBC Drivers	1390
Specifying Exclusions for Deployment Testing	1390
Lockdown and Runtime Testing	1392
Performing Lockdown and Runtime Tests	1393
Performing Lockdown and Runtime Tests Under a Different User Account	1394
Running Lockdown and Runtime Tests in Restricted Environments	1395
Performing Isolation Tests	1395
Filtering Results of Lockdown and Runtime Tests	1396
Using MSI Doctor to Verify Package Deployment Status	1397
View Product, Feature, or Component Deployment Status Properties	1399
Verify Product, Feature, or Component Data	1399
Install or Configure Products or Features	1400
Reinstall Features	1401
Reinstall Components	1402
Creating Custom Test Cases	1402
Test Reports	1404
Running QualityMonitor from the Command Line	1405
QualityMonitor Reference	1406
Menus and Toolbar	1406
QualityMonitor Interface	1408
Dialog Boxes	1408
About QualityMonitor Dialog Box	1409
Add Exclusions Dialog Box	1409
Component Properties Dialog Box	1409

Feature Properties Dialog Box	1410
Install or Configure Feature Dialog Box	1411
Install or Configure Product Dialog Box	1411
Installed Data Dialog Box	1412
Open QualityMonitor Project Dialog Box	1413
Options Dialog Box	1414
<i>General Tab</i>	1414
<i>Exclusions Tab</i>	1415
Product Properties Dialog Box	1416
Re-install Product/Feature Dialog Box	1417
Runtime Test Filters Dialog Box	1417
Test Item Information Dialog Box	1418
Test Progress Dialog Box	1418
Test Result Dialog Box	1418
Views	1418
Welcome to QualityMonitor View	1419
Product Information View	1420
Test Cycle Summary View	1420
Deployment Tests View	1420
Class IDs View	1421
File Associations View	1422
Help Files View	1422
Prog IDs View	1422
Shortcuts View	1422
Type Libraries View	1422
Manifests View	1423
ODBC Data Sources View	1424
ODBC Drivers View	1424
Services View	1424
Lockdown and Runtime Tests View	1425
Runtime Execution Details View	1425
Files View	1426
Folders View	1427
Registry Entries View	1429
Isolation Tests View	1430
User-Defined Tests View	1431
Test Case View	1431
Deployment Status View	1431

Part 5: Preparing Installations for Deployment. 1433

18 Preparing Packages for Distribution Using Distribution Wizard 1435

Creating Administrative Installations for Packages 1436

Distributing Packages to FTP Servers	1437
Preparing for Altiris Distribution	1437
Preparing for LANDesk Distribution.....	1439
Preparing for ManageSoft Distribution	1439
Using Marimba Channel Publishing for Package Deployment.....	1441
Distributing Packages to Network Locations	1443
Preparing for SMS Distribution.....	1444
SMS File Templates	1445
Preparing for Tivoli Distribution	1446
Preparing for ZENworks Configuration Management 10 Distribution.....	1447
Preparing for ZENworks Desktop Application Distribution	1452
Preparing for ZENworks Server Distribution.....	1456
Deploying InstallScript MSI Installations.....	1461
Reference	1462
Distribution Wizard	1463
Welcome Panel	1463
Distribution Type Panel.....	1464
Administrative Install Panel.....	1465
Altiris Integration Panel	1465
Altiris XML Template	1466
FTP Location Panel.....	1467
LANDesk Integration Panel.....	1467
ManageSoft Distribution Settings Panel	1468
ManageSoft Package and Environmental Settings Panel.....	1468
Marimba Panels	1469
Patches Panel.....	1469
Marimba Integration Panel	1470
Advanced Marimba Options Dialog Box	1471
Network Location Panel	1471
SMS Distribution Panel.....	1472
Tivoli Integration Panel	1472
Tivoli Settings Panel	1472
SPD Parameters Panel	1473
SPD File Details	1474
ZENworks Login Panel	1477
ZENworks Desktop Application Panel	1480
ZENworks Server Distribution/Object Panel	1482
ZENworks Server Distribution/Distributor Panel.....	1483
Installation Package Information Panel	1485
Distribution Summary Panel.....	1486
Distribution Output Panel	1486
Distribution Wizard for ZENworks Configuration Management.....	1486

Welcome Panel	1487
Login Panel	1487
Windows Installer Package Information Panel	1488
Bundle Creation Options Panel	1494
Bundle Information Panel	1494
Summary Panel	1495
Publishing Process Panel	1496

19 Distributing Packages Using Configuration Manager Web Console 1497

About Microsoft System Center Configuration Manager and Configuration Manager Web Console 1498

Distributing Software Through Configuration Manager Server	1499
Configuration Manager Web Console Overview	1501
Getting Started	1502

Distributing Packages Using the Configuration Manager Web Console 1502

Configuring Distribution Settings	1503
Distributing a New Package	1503
Selecting a Package for Distribution	1504
Setting Package Configuration Options	1505
<i>Setting Package Settings Options</i>	1506
<i>Setting Access Accounts Options</i>	1510
<i>Setting Distribution Points Options</i>	1511
<i>Setting Programs Options</i>	1512
<i>Setting Advertisements Options</i>	1519
Saving the Package on the Configuration Manager Server	1522
Viewing the Status of a Distributed Package	1522
Modifying the Distribution Settings of a Package	1522
Deleting Packages	1523

Configuration Manager Web Console Reference 1525

Microsoft System Center Configuration Manager Web Console Home Page	1526
Connect to Configuration Manager Page	1526
Distribute Package Page	1527
PDF Selection Page	1527
Configuration Manager Package Configuration Page	1528
<i>Package Settings View</i>	1529
<i>Access Accounts View</i>	1532
<i>Distribution Points View</i>	1533
<i>Programs View</i>	1534
<i>Advertisements View</i>	1541
Package Summary Page	1544
Select a Configuration Manager Collection Dialog Box	1544
Distribution Settings Page	1544
Package Administration Page	1545
Package Status Page	1545
Confirm Delete Page	1548

Distribute New Package Page	1549
-----------------------------------	------

20 Analyzing Installations Prior to Deployment 1551

About the Predeployment Test Tool.....1552

How the Predeployment Test Tool Works	1552
Predeployment Test Tool Components	1552
Overview of Predeployment Test Tool Results	1553

Configuring the Predeployment Test Environment.....1554

Configuring the Predeployment Test Results Web Site.....	1554
Configuring the Predeployment Test Web Service	1556

Performing Predeployment Analysis1558

Creating a Test-Ready Windows Installer Package	1558
Setting Predeployment Test Command Line Parameters	1560
Distributing a Test-Ready Windows Installer Package	1562
Running Diagnostic Tests When Distributing Test-Ready Packages	1562

Viewing Predeployment Test Results.....1564

Managing Job Data	1566
-------------------------	------

Predeployment Test Reference1567

Wizards and Dialog Boxes	1568
Predeployment Test Start Page	1568
Predeployment Test Preparation Wizard	1569
<i>Welcome Panel</i>	<i>1569</i>
<i>Windows Installer File Selection Panel.....</i>	<i>1570</i>
<i>Web Service URL Panel</i>	<i>1571</i>
<i>Select Application Catalog Panel</i>	<i>1571</i>
<i>Deployment & Conflict Test Selection Panel</i>	<i>1572</i>
<i>Review Panel</i>	<i>1573</i>
<i>Progress Panel</i>	<i>1574</i>
<i>Summary Panel.....</i>	<i>1574</i>
Options Dialog Box.....	1574
Configure Application Catalog for Predeployment Web Service Page	1575
Technical Information	1575
Predeployment Custom Properties	1576
Predeployment Test Identifiers.....	1576
Predeployment Test Results XML File	1577
Changes Made During Test-Ready Package Creation	1582
Predeployment Test Results.....	1583
Predeployment Test Results By Job Report	1584
Job Summary Report	1585
Test Result Summary Report	1586
Job Details Report	1588
Test Details Report	1589
File/Registry Conflicts Report	1591

Predeployment Test Results By Machine Report	1593
Machine Result Summary Report	1593
Connection to New Catalog Page	1595
Manage Job Page	1596
Assigning ASP.NET Permissions to Files or Folders	1597
ConflictSolver ACE Reference	1601
ACE07	1602
ACE10	1603
ACE23	1605
ACE24	1607

21 License-Enabling Packages Using FLEXwrap 1609

Enabling License Tracking of Windows Installer Packages 1610

About FLEXnet Licensing Technology and FLEXwrap	1610
Benefits of FLEXnet Software Licensing Technology	1611
Types of Applications that Can Be FLEX-enabled	1612
About FLEXwrap License Files	1612
Selecting the Executable File to Wrap	1613
About Output Folders	1613
How FLEXwrap Assistant Works	1614
Launch Process for a Wrapped Application	1616
Using the FLEXwrap Assistant	1618
Launching the FLEXwrap Assistant	1618
License-Enabling a Windows Installer Package	1619
<i>Selecting a Windows Installer Package to License-Enable</i>	<i>1619</i>
<i>Setting License Limits and Queuing Behavior</i>	<i>1620</i>
<i>Defining License Requirements and Access Restrictions</i>	<i>1621</i>
<i>Configuring a Connection to the FLEXwrap License Server</i>	<i>1622</i>
<i>Specify the Output Folder and Build Package</i>	<i>1624</i>
<i>Next Steps</i>	<i>1624</i>
Setting Licensing Options	1624
<i>Benefits of Using the Suite Option</i>	<i>1625</i>
<i>Benefits of Encrypting the Source Executables</i>	<i>1625</i>
<i>Setting Your License Limit</i>	<i>1625</i>
<i>Using Overdraft Support to Determine Application Usage</i>	<i>1626</i>
<i>Setting a License Expiration Date</i>	<i>1626</i>
<i>Choosing the Right Queuing Option for Your Application</i>	<i>1626</i>
<i>Grouping Options for Concurrent Use</i>	<i>1627</i>
<i>Setting Licensing Behavior During Server Down Time</i>	<i>1627</i>
<i>Setting Timeout Periods</i>	<i>1627</i>
Setting Configuration Options	1628
<i>About Hostids</i>	<i>1628</i>
<i>Changing the Host Name or Port at Install Time</i>	<i>1628</i>
<i>When to Update the Package Code</i>	<i>1628</i>

<i>FLEX-enabling a Signed Package</i>	1629
<i>Purpose of Packaging Comments</i>	1629
FLEXwrap Assistant Reference	1629
Home Page	1630
Select Files Page	1631
Set Limits Page	1632
Define Access Page	1633
Configure Connection Page	1634
Finalize Package Page	1635
Permitted Hostid Dialog Box	1636
Options Dialog Box	1636
FLEXwrap Server Name Dialog Box	1637
Output Window	1637
Managing FLEXwrap License Files	1637
Overview of the FLEXwrap License Server	1638
Supported Platforms	1639
Communications Issues	1640
Installation Notes	1640
About the FLEXwrap Server Configuration Tool	1641
About the FLEXwrap LMTOOLS	1641
<i>Configuring License Server Using a License File</i>	1642
<i>Configuring License Server Using Services</i>	1642
Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server	1642
Installing Your Production FLEXwrap License File	1642
Starting the License Server	1645
<i>Configuring FLEXwrap License Server to Run as a Service</i>	1645
<i>Manually Starting the FLEXwrap License Server from the FLEXwrap Interface</i>	1647
<i>Manually Starting the FLEXwrap License Server from the Command Line</i>	1648
Stopping the License Server	1649
Checking the Status of the FLEXwrap License Server	1649
<i>Checking the License Server Status from the FLEXwrap Server Configuration Tool</i>	1649
<i>Checking the License Server Status from the LMTOOLS Interface</i>	1650
Tracking Projects with FLEXwrap	1651
Using the FLEXwrap Server Configuration Tool to FLEX-Enable Executable (.exe) Files	1651
Overview	1652
Available Options	1654
FLEX-Enabling an Executable (.exe) File	1654
<i>Specifying Basic Settings for the License</i>	1655
<i>Specifying Advanced Settings for the License and Binary</i>	1656
<i>Creating an Options File</i>	1660
<i>Saving the New License File and Wrapped Application</i>	1661
<i>Testing a Wrapped Application</i>	1662
Setting Up Fail Safe Modes	1663
<i>Setting Up Regular (License) Failsafe Mode</i>	1663

<i>Setting Up License Server Failsafe Mode</i>	1664
FLEXwrap Server Configuration Tool Reference	1666
FLEXwrap Terminology	1666
About FLEXwrap License Parameters	1668
<i>Specifying FLEXwrap License Parameters</i>	1668
<i>Understanding FLEXwrap License File FEATURE Lines</i>	1673
Dialog Boxes and Wizards	1675
<i>FLEXwrap Server Configuration Dialog Box</i>	1676
<i>FLEXwrap Application Information Dialog Box</i>	1680
<i>Wrapping Wizard</i>	1685
<i>Encryption Wizard</i>	1687
<i>Wrapping and Encryption Wizard</i>	1688
<i>FLEXwrap Command Dialog Box</i>	1691
<i>Invalid Certificate Dialog Box</i>	1692

Part 6: Using AdminStudio Enterprise Server Tools 1693

22 Generating and Viewing Reports in Report Center 1695

Generating and Viewing AdminStudio Reports	1695
About the Package Report Tab of the Report Center	1697
Generating a Package Report	1700
Viewing Package Reports	1700
Searching for a Package on the Select Package Page	1700
Information Included in Package Reports	1704
Package Summary Information View	1705
Files View	1706
Registry View	1707
Shortcuts View	1708
ODBC Drivers View	1709
ODBC DS View	1710
Extended Attributes View	1711
Validation View	1711
Conflicts View	1713
History View	1714
Dependencies View	1715
Navigating Through a Package Report	1716
Archiving a Package Report	1718
Exporting a Package Report	1719
Generating a Custom SQL Query Report for AdminStudio	1720
Report Center Reference	1721
All Reports Page	1722
Report View	1722
Select Package Page	1723

Contents

Package Report	1728
Reports Wizard	1733
Select Report Type Panel	1733
Enter SQL Query Panel	1733
General Information Panel	1734
Summary Panel	1734
AdminStudio and Workflow Manager Tables	1735
AdminStudio Tables	1735
cstblPackage	1735
cstblGroups	1737
cstblGroupPackages	1737
AsJob	1738
AsJobQueue	1738
AsJobStep	1739
Workflow Manager Tables	1740
Workflow Manager Database Table Schema Diagram	1741
AMS_Application Table	1743
AMS_ApplicationItem Table	1744
AMS_Person Table	1745
AMS_ApplicationStatus Table	1746
AMS_RelatedApplications Table	1746
AMS_Issues Table	1746
AMS_AuditTrail Table	1748
AMS_Contracts Table	1751

23 Automating Tasks Using Job Manager..... 1753

About Job Manager	1754
Job Steps: Tasks You Can Automate	1755
Template Jobs vs. Custom Jobs	1756
Selecting Contextual Packages in a Custom Job	1757
Dynamic Group Content Identification	1757
Conflict Resolution and the Software Repository	1758
User Permissions and Job Manager	1759
Directory Monitoring and Job Manager	1759
Managing Jobs	1760
Creating a New Job	1760
Creating a New Job Based on a Template	1760
Creating a New Custom Job	1761
Editing an Existing Job	1766
Rescheduling a Job	1767
Setting Email Alerts	1768
Viewing Job Status	1769
Deleting a Job	1770

Managing Templates	1771
Source and Target Packages in Template Jobs vs. Custom Jobs	1771
Creating a New Template	1772
Editing an Existing Template	1774
Deleting a Template	1775
Setting Job Manager Conflict Detection Options	1775
Managing Jobs Using Job Manager Engine	1777
About Job Manager Engine	1777
Opening the Job Manager Engine User Interface	1778
Adding an Application Catalog	1778
Deleting a Connection to an Application Catalog	1778
Executing Jobs	1779
Pausing, Shutting Down, and Restarting Application Catalog Jobs	1779
Viewing Job Manager Engine Status in Job Manager	1780
Setting Configuration Options	1781
Job Manager Reference	1783
Jobs Queue Page	1784
Add Job Page	1785
Edit Job Page	1788
Delete Job/Template Page	1790
Schedule Job Page	1790
Job Alerts Page	1791
Job Status Page	1791
Job Template List Page	1792
Add Template Page	1792
Edit Template Page	1794
Conflict Detection Options Page	1795
Job Manager Engine	1796
Manage Application Catalogs View	1796
Conflict Detection Configuration View	1798
Validation Configuration View	1799
E-mail Alerts View	1799
24 AdminStudio.Public API Reference	1801
Index	1803

Part 1

Getting Started

This part of the AdminStudio 9.5 User Guide includes the following chapters:

- [AdminStudio 9.5 Help Library](#)
- [Using the Process Assistant](#)
- [Using the AdminStudio Interface](#)

AdminStudio 9.5 Help Library

AdminStudio enables systems administrators to rapidly prepare error-free applications to deploy into their enterprise environment through a structured process built on application management best practices.

The [AdminStudio Start Page](#) provides quick access to AdminStudio tools, product information, recently opened files, and InstallShield resources.

The AdminStudio user documentation contains information about the functionality and features of all of the components of AdminStudio and is presented in the following sections:

Table 1-1 • AdminStudio Help Library




	Topic	Content
Part 1: Getting Started		
	Using the Process Assistant	Explains how to use the Process Assistant, a procedure-based alternative interface to the standard AdminStudio interface.
	Using the AdminStudio Interface	Describes the AdminStudio Interface, the central application for AdminStudio. From it, you can launch the AdminStudio tools, create workflow templates and projects, use AdminStudio Enterprise Server tools, and connect to and create Application Catalogs.
Part 2: Managing Users, Roles, Permissions, Application Catalogs		
	Managing Users, Directory Services, and User Logins	Explains how to create an account for each person that you want to have access to AdminStudio, and how to import users or groups of users from a directory service. Also explains how to set up the AdminStudio account, domain account, single sign-on, and guest account login methods.

Table 1-1 • AdminStudio Help Library (cont.)










	Topic	Content
	Managing Roles and Permissions	Explains how to create and edit roles to manage access to AdminStudio functionality.
	Managing Application Catalog Databases	Explains how to use Application Manager to import applications into the Application Catalog, organize them, and set up automatic import. It also explains how to share Application Catalog data in your enterprise.
Part 3: Repackaging and Customizing Installations		
	Repackaging Legacy Installations Using the Repackaging Wizard	Explains how to use Repackager's Repackaging Wizard to convert existing legacy installations into Windows Installer (MSI) packages.
	Converting Legacy Installations Using the Repackager Interface	Explains how to use the Repackager interface to create and modify Repackager project files, and how to build those files into InstallShield Editor projects or Windows Installer packages.
	Converting Windows Installer Packages to Virtual Applications	Explains how to use Repackager's Windows Installer Converter to create virtual applications.
	Creating Customized Virtual Applications	Explains how to use the InstallShield Virtualization Assistants to create customized virtual applications in the Microsoft App-V, VMware ThinApp, and Citrix XenApp virtual application formats.
	Customizing and Authoring Installations Using InstallShield	Describes how to use InstallShield Editor to create setup packages that utilize Windows Installer technology, while harnessing the flexibility provided by InstallScript. Also explains how to use InstallShield to create virtual applications.
	Customizing Installations Using Tuner	Explains how to use Tuner to create a transform file to add to, modify, or remove information from a Windows Installer package.
Part 4: Testing Windows Installer Packages		
	Identifying and Resolving Application Conflicts Using ConflictSolver	Explains how to use ConflictSolver to identify conflicts between Windows Installer packages before you deploy them, and resolve the problems before they affect your end users.

Table 1-1 • AdminStudio Help Library (cont.)











	Topic	Content
	Identifying and Resolving Package Errors Using PackageExpert	Explains how to use PackageExpert to automate the operation of specific tests and resolutions against Windows Installer (.msi) software packages. PackageExpert can check the integrity of your Windows Installer packages, and identify and resolve any issues relating to Windows Vista OS deployment.
	Analyzing the Impact of Installing Microsoft Operating System Patches	Explains how to import Microsoft application patches into the Application Catalog and thoroughly test the impact they will have on your environment before they are deployed.
	Isolating Applications Using Application Isolation Wizard	Explains how to use the Application Isolation Wizard to solve component versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.
	Ensuring Package Quality Using QualityMonitor	Explains how to use QualityMonitor to run a series of built-in tests to installed Windows Installer–based products, helping to ensure they run correctly, especially in a locked-down environment.
Part 5: Preparing Installations for Deployment		
	Preparing Packages for Distribution Using Distribution Wizard	Explains how to use the Distribution Wizard to assist you in deploying your package (and any associated transforms) to a network location, an FTP server, as an administrative installation, or using any of the following distribution technologies: <ul style="list-style-type: none"> • Altiris • LANDesk • ManageSoft • Marimba • Microsoft Configuration Manager • Microsoft SMS • Tivoli • ZENworks
	Distributing Packages Using Configuration Manager Web Console	Explains how to use Configuration Manager Web Console , a Web-based application, to configure package distribution options and distribute setup packages without going to Configuration Manager Server.

Table 1-1 • AdminStudio Help Library (cont.)

	Topic	Content
	Analyzing Installations Prior to Deployment	Explains how to use the Predeployment Test tool to determine if a Windows Installer .msi package will succeed or fail when it is installed in production by finding issues with disk space, install conditions, and conflicts with .msi packages already installed on the system.
	License-Enabling Packages Using FLEXwrap	Explains how to use FLEXwrap to inject software licensing technology into existing Windows Installer (.msi) packages so that they can track usage of the executables in those packages.
Part 6: Using Enterprise Tools: Job Manager and Report Center		
	Generating and Viewing Reports in Report Center	Explains how to use the Report Center to report on or view all of the information regarding the applications in your Application Catalog from a single location.
	Automating Tasks Using Job Manager	Explains how to use Job Manager to automate time consuming application migration tasks, saving you time and enabling you to enforce standardized business practices on the packaging process.



Note • In October 2009, Acreesso Software changed its name to Flexera Software, Inc. AdminStudio 9.5 includes the new name Flexera Software. The AdminStudio Editor features functionality from InstallShield 2010 Service Pack 1. Because InstallShield 2010 Service Pack 1 was released prior to the name change, it still contains references to Acreesso Software. In future releases, all logos and trademarks will be Flexera Software. The current logos and trademarks cause no issues for users of AdminStudio 9.5.

What's New in AdminStudio 9.5

This section describes the new features in the following versions of AdminStudio:

- [AdminStudio 9.5 SP1](#)
- [AdminStudio 9.5](#)

AdminStudio 9.5 SP1

The AdminStudio 9.5 SP1 release includes the following new features:

- [Repackaging 64-Bit Applications](#)
- [Creating 64-Bit Microsoft App-V Applications](#)
- [PackageExpert Windows 7 Compatibility Tests](#)

Repackaging 64-Bit Applications

Repackager now has the capability to repackaging both 32-bit and 64-bit applications, as well as hybrid applications (both 32-bit and 64-bit). The Repackaging Wizard remains a 32-bit application, but can be run on both 32-bit (x86) and 64-bit (x64) Windows operating systems. For more information, see [Repackaging 64-Bit Applications](#).

Creating 64-Bit Microsoft App-V Applications

AdminStudio now supports converting a 64-bit Windows Installer package into a 64-bit Microsoft App-V package, which can be deployed on Windows 64-bit systems with the Microsoft App-V 4.6 64-bit client. This, in combination with our new 64-bit repackaging capability, enables you to convert any 64-bit installation into a 64-bit App-V package. For more information, see [Creating 64-Bit App-V Packages](#).

PackageExpert Windows 7 Compatibility Tests

PackageExpert now includes Windows 7 compatibility tests. Windows 7 compatibility tests find and resolve issues packages would have when deployed on the Windows 7 operating system. The following Windows 7 tests were added:

Table 1-2 • PackageExpert Windows 7 Compatibility Tests

Test	Description
Custom Action Impersonation Test 1	Identifies custom actions that are not scheduled to run.
Custom Action Impersonation Test 2	Identifies scheduled custom actions with the wrong impersonation setting.

Table 1-2 • PackageExpert Windows 7 Compatibility Tests

Test	Description
Null Component ID	Checks for missing component IDs.
Nested Custom Actions	Checks for nested install custom actions.
Restart Manager Files in Use Dialog	Checks for the presence of a MsiRMFilesInUse dialog.
Reboot Pending Launch Condition	Checks for a launch condition that will prevent an installation from continuing when a restart is pending.
Per User Setups	Checks components for mixed per-user and per-machine data.
ForceReboot	Identifies the use of the ForceReboot action.
AdminUser in InstallUISequence	Identifies the use of the AdminUser property in the InstallUISequence table.
AdminUser or Privileged Launch Condition	Identifies the use of the AdminUser or Privileged properties in the LaunchCondition table.
Windows 7 Protected Files	Verifies that your installation does not install files that are protected by Windows Resource Protection (WRP).
Windows 7 Protected Registry	Verifies that your installation does not install registry keys that are protected by Windows Resource Protection (WRP).

AdminStudio 9.5

AdminStudio 9.5 includes the following new features:

- [InstallShield 2010 SP1](#)
- [Virtualization Pack Enhancements](#)
- [Repackager Enhancements](#)

InstallShield 2010 SP1

AdminStudio 9.5 now includes InstallShield 2010 SP1, which offers many new features and enhancements that will help you use the latest technologies and save time on everyday tasks. InstallShield 2010 SP1 includes the following new features and enhancements:

- [Ability to Target Windows 7 and Windows Server 2008 R2 Systems](#)
- [Microsoft SQL Server 2008 SP1 Support](#)
- [Windows Installer 5 Support](#)

- Improved User Interface
- New String Editor View
- New View for Configuring Run-Time Text File Changes
- New Tool for Scanning an IIS Web Site, Recording Its Settings, and Importing Those Settings into an InstallShield Project
- Ability to Add IIS Web Applications to Web Sites, Plus Other IIS-Related Improvements
- Support for InstallShield Prerequisites in InstallScript Projects
- Ability to Detect the Presence of a Virtual Machine
- New Support for Setting Permissions for Files, Folders, and Registry Keys
- Unicode Support
- Support for 64-Bit COM Extraction
- New InstallShield Prerequisites
- New Billboard Support in Basic MSI Projects
- Expanded Billboard Support in InstallScript and InstallScript MSI Projects
- Support for HTML Controls on Dialogs During InstallScript and InstallScript MSI Installations
- Support for Specifying Action Progress Messages
- Setup.exe Manifests Now Have Compatibility Section to Avoid Triggering Program Compatibility Assistant on Windows 7 and Windows Server 2008 R2 Systems
- Updated Visual Studio Merge Modules
- Windows Installer 5 Support for Configuring New Customization Options for Windows Services, Plus Other Service-Related Improvements
- Validation for the Windows 7 Logo Program
- Windows Installer 5 Support for Setting Windows Shell Properties for a Shortcut
- Ability to Specify a Custom .png Shortcut Icon for the Start Screen on Windows Mobile 6.5 Professional-Powered Devices
- Windows Installer 5 Support for the MsiPrint and MsiLaunchApp Control Events
- Windows Installer 4.5 Prerequisites for Windows Vista Systems
- Dependency Scanning and COM Extraction Now Exclude Protected Windows 7 Files and Registry Entries

Ability to Target Windows 7 and Windows Server 2008 R2 Systems

InstallShield now enables you to specify that your installation requires Windows 7 or Windows Server 2008 R2. It also lets you build feature and component conditions for these operating systems.

Microsoft SQL Server 2008 SP1 Support

InstallShield now includes support for running SQL script on SQL Server 2008 SP1. In addition, InstallShield includes SQL Server 2008 SP1 in the predefined list of database servers that you can select when you are specifying in the SQL Scripts view the target database servers that your product supports.

Windows Installer 5 Support

Windows Installer 5 is available with the releases of Windows 7 and Windows Server 2008 R2. With InstallShield 2010's own implementation of Windows Installer 5 features, you can take advantage of the new Windows Installer 5 functionality, including:

- **Support for Per-User Installations**—The General Information view has a new **Show Per-User Option** setting. This setting lets you specify whether you want the **ReadyToInstall** dialog—in certain scenarios—to include buttons that let end users indicate how they want to install the product: for the current user or for all users. The per-user button sets the new Windows Installer property **MSIINSTALLPERUSER** equal to 1 to indicate that the package should be installed for the current user.
- **Support for Reducing the Time for Installing Large Packages**—Use the new **Fast Install** setting in the General Information view to select one or more options that may help reduce the time that is required to install a large Windows Installer package. For example, you can specify that you do not want a system restore point to be saved for your installation. You can also specify that you want the installation to perform file costing, but not any other costing. This setting configures the new Windows Installer property **MSIFASTINSTALL**, which can be set at the command line.
- **Support for Additional Windows Installer Features**—The Director Editor in InstallShield includes support for the new Windows Installer 5 tables (**MsiLockPermissionEx**, **MsiServiceConfig**, **MsiServiceConfigFailureActions**, and **MsiShortcutProperty**).

Improved User Interface

InstallShield user interface enhancements include new toolbars and views, customizable grids, and dynamic search tools. Highlights include:

- **Direct Editor view**—When you select a table in this view, a new toolbar is displayed. The toolbar has buttons that let you add, cut, copy, paste, find, and replace data in the table. This view also has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the table records that do not contain it. When you are adding or editing a record, InstallShield displays help text for each field.
- **Property Manager view**—This view contains a new toolbar that has buttons for performing tasks such as adding and deleting properties. This view also has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the table records that do not contain it. The top of the view has a new group box area; you can drag and drop column headings onto this area to organize the rows in the view in a hierarchical format. You can now select multiple properties in this view (by using the mouse and the SHIFT or CTRL button) and then delete them all at once.
- **Redistributables view**—The new toolbar and the new group box area in this view provide robust search and organizational functionality. You can drag and drop column headings onto the group box area to organize the list of redistributables in a hierarchical format. In addition, you can type a string in the toolbar's search box, and InstallShield hides all of the redistributables that do not contain it.

- **Internet Information Services view**—This view has been redesigned to look similar to IIS 7: the settings are now displayed in grids, instead of on tabs. The grids have buttons that let you sort the grid settings by category or alphabetically. When you select a setting in one of the grids in this view, InstallShield displays help information for that setting in the lower-right pane.
- **General Information view**—All of the settings in this view are displayed in one grid, instead of as separate grids that are associated with nodes within this view. The settings are grouped into several categories to make it easy to find a particular setting. In addition, the grid has a button that lets you sort the grid settings by category or alphabetically. The string tables that were previously in this view have been moved to a new [String Editor](#) view.
- **Path Variables view**—This view contains a new toolbar that has buttons for performing tasks such as adding and deleting path variables. This view also has support for dynamic searches—as you are typing a string in the search box, InstallShield hides all of the rows that do not contain it. The top of the view has a new group box area; you can drag and drop column headings onto this area to organize the rows in the view in a hierarchical format.

New String Editor View

InstallShield has a new String Editor view. This view contains a spreadsheetlike table that shows the collection of language-independent identifiers and corresponding language-specific values for your project. In the String Editor view, you have complete and centralized control over the localizable text strings that are displayed at run time during the installation process. The view replaces the string tables that were previously available as nodes within the General Information view.

New View for Configuring Run-Time Text File Changes

InstallShield has a new Text File Changes view, which enables you to configure search-and-replace behavior for content in text files—for example, .txt, .htm, .xml, .config, .ini, and .sql files—that you want to modify at run time on the target system. The text files can be part of your installation, or they can be files that are already present on target systems.

You can use Windows Installer properties to specify the names of the text files that you want to include in or exclude from your search. You can also use properties to specify the search strings and the replacement strings. This enables you to use data that end users enter in dialogs, or other configuration information that is determined at run time, when your product's text files are modified on the target system. For example, if your project includes a dialog in which end users must specify an IP address, your installation can search a set of files for a token, and replace it with the IP address that end users enter.

New Tool for Scanning an IIS Web Site, Recording Its Settings, and Importing Those Settings into an InstallShield Project

InstallShield includes an IIS scanner (IISscan.exe), a new command-line tool for scanning an existing IIS Web site and recording IIS data about the Web site. The IIS scanner creates an XML file that contains all of the settings for the Web site, its virtual directories, its applications, and its application pools. You can use the XML file to import the IIS data into the Internet Information Services view in InstallShield. Once you have imported the IIS data into a project, you can use the Internet Information Services view to make changes to the IIS settings as needed.

Ability to Add IIS Web Applications to Web Sites, Plus Other IIS-Related Improvements

InstallShield now lets you add IIS Web applications to Web sites. You can do so by right-clicking a Web site in the Internet Information Services view and clicking New Application. Once you have added a new application, you can configure its settings in the right pane.

InstallShield also lets you create a virtual directory without an application. Previously whenever you created a virtual directory, an application was also created automatically.

In addition, InstallShield has new IIS settings:

- **Managed Pipeline Mode**—This setting enables you to specify the appropriate request-processing pipeline mode—either integrated or classic—for an application pool.
- **Enable 32-Bit Applications**—This setting lets you specify whether you want to allow 32-bit applications in the selected application pool to be run on 64-bit systems.
- **.NET Framework Version**—This setting is where you specify the version of the .NET Framework that an application pool should load.
- **ASP.NET Platform**—If your installation may be run on a 64-bit version of Windows with the .NET Framework, use this setting to specify which ASP.NET platform should be used to map a Web site, application, or virtual directory to the ASP.NET version.

Support for InstallShield Prerequisites in InstallScript Projects

InstallShield now enables you to add InstallShield prerequisites to InstallScript projects. Previously, only Basic MSI and InstallScript MSI projects included support for this type of redistributable.

InstallShield prerequisites are redistributables that usually install a product or technology framework that is required by your product. Now you can add any of the InstallShield prerequisites that are provided with InstallShield—or any of your own custom-designed InstallShield prerequisites—to InstallScript projects. If you work on a mix of different project types, InstallShield lets you simplify your testing matrix by enabling you to reuse this type of redistributable in all of your Basic MSI, InstallScript, and InstallScript MSI projects.

Ability to Detect the Presence of a Virtual Machine

InstallShield lets you determine whether an installation is running on any of the following types of virtual machines:

- Microsoft Hyper-V
- A VMware product such as VMware Player, VMware Workstation, or VMware Server
- Microsoft Virtual PC

Two new Windows Installer properties are available when you add an MSI DLL custom action to your project to detect virtual machines: **IS_VM_DETECTED** and **IS_VM_TYPE**. The custom action should be configured to call the **ISDetectVM** function in the SetAllUsers.dll file, which is installed with InstallShield.

In addition, the InstallScript language has been expanded to support the detection. The structure **SYSINFO** contains a new **bIsVirtualMachine** member, and a new **VIRTUAL_MACHINE_TYPE** constant is available for use with the InstallScript function **GetSystemInfo**.

New Support for Setting Permissions for Files, Folders, and Registry Keys

InstallShield offers two new ways to secure files, folders, and registry keys for end users who run your product in a locked-down environment:

- **Custom InstallShield handling**—In Windows Installer-based projects, you can choose to use custom support for setting permissions at run time. With this option, InstallShield stores permission information for your product in the custom **ISLockPermissions** table of the .msi database. InstallShield also adds custom actions to your project to set the permissions.
- **SetObjectPermissions, an InstallScript Function**—You can use the new **SetObjectPermissions** function in InstallScript events and InstallScript custom actions to set permissions at run time.

With the custom InstallShield handling option, the file, folder, or registry key for which you are setting permissions must be installed as part of your installation. With the **SetObjectPermissions** function, the file, folder, or registry key can be installed as part of your installation, or it can be already present on the target system.

Previously, the only option that InstallShield offered for setting permissions was to use the traditional Windows Installer handling. With this option, the permission information is stored in the **LockPermissions** table of the .msi database. The new custom InstallShield handling option and the **SetObjectPermissions** function offer several advantages over the traditional Windows Installer handling:

- The custom option and the **SetObjectPermissions** function include support for many well-known security identifiers (SIDs) that are not supported by the traditional Windows Installer handling option.
- The custom option and the **SetObjectPermissions** function support the use of localized user names for the supported SIDs, unlike the traditional option. With the traditional option, if you try to use a localized name to set permissions on a non-English system, the installation may fail.
- The custom option and the **SetObjectPermissions** function let you specify that you want to deny a user or group from having the permissions that you are specifying. The traditional handling does not allow you to do this.
- The custom option and the **SetObjectPermissions** function let you add permissions to a file, folder, or registry key that already exists on the target system, without deleting any existing permissions for that object. With the traditional handling, the existing permissions are deleted.
- The custom option and the **SetObjectPermissions** function let you configure permissions for a folder (or a registry key), and indicate whether you want the permissions to be applied to all of the folder's subfolders and files (or the registry key's subkeys). With the traditional handling, if you want to configure permissions for a subfolder or a file in a folder (or a subkey under a registry key), the parent that is created on the target system automatically inherits the permissions of its child.
- The custom option and the **SetObjectPermissions** function let you configure permissions for a new user that is being created during the installation. The traditional handling does not allow you to do this; the user must already exist on the target system at run time.

The General Information view has a new **Locked-Down Permissions** setting that lets you specify whether you want to use the new custom InstallShield handling or the traditional Windows Installer handling for all new permissions that you set for files, folders, and registry keys in your project. If you have already configured some permissions in your project and then you change the value of this setting, InstallShield lets you specify whether you

want to use the alternate handling method for those already-existing permissions. In all new projects, the default value for this setting is the custom InstallShield handling option. If you upgrade a project from InstallShield 2009 or earlier to InstallShield 2010, the traditional Windows Installer handling option is the default value of this setting.

Unicode Support

Windows Installer databases can now be built in a Unicode format, InstallShield projects are now stored in a Unicode format, and the InstallShield interface now supports entering and viewing Unicode characters from multiple character sets at the same time.

Support for 64-Bit COM Extraction

InstallShield now supports 64-bit COM extraction. If you are using InstallShield on a 64-bit operating system, InstallShield can extract COM data from a 64-bit COM server. In order to install the data to the correct locations, the component must be marked as 64 bit.

New InstallShield Prerequisites

InstallShield includes a number of new InstallShield prerequisites that you can add to your projects. New InstallShield prerequisites include:

- Windows Installer 4.5 (with Microsoft fix), 3.1, 3.0, and 2.0
- .NET Framework 3.0 SP1 and 2.0 SP2
- Internet Explorer 8
- Microsoft SQL Server 2008 Express SP1 and 2005 Express SP3
- Microsoft Visual C++ 2005 SP1
- Oracle 11g Instant Client 11.1.0.7
- Crystal Reports Basic for Visual Studio 2008

New Billboard Support in Basic MSI Projects

Basic MSI projects now include support for billboards. You can add billboards to your projects to display information to end users during the installation process. The billboards can be used to communicate, advertise, educate, and entertain end users. For example, billboards can present overviews on new features of the product being installed or other products from your company. Each billboard is a file that you or your company's graphics department creates for complete control over the look and feel of the file transfer.

Following are some of the highlights regarding billboard support in Basic MSI projects:

- You can add an Adobe Flash application file (.swf) as a billboard in your project.
- Flash application files can consist of videos, movies, sounds, interactive interfaces, games, text, and more—anything that is supported by the .swf type of file.
- InstallShield lets you use .bmp, .gif, .jpg, and .jpeg files as billboards.
- InstallShield includes a Billboard Type setting that lets you specify which style of billboard you want to use in your installation. For example, with one style, the installation displays a full-screen background, with billboards in the foreground, and a small progress box in the lower-right corner of the screen. With another style, the

installation displays a standard-size dialog that shows the billboards. The bottom of this dialog shows the progress bar.

- InstallShield lets you preview a billboard to see how it would be displayed at run time, without requiring you to build and run a release. Previewing a billboard lets you see how your billboard will look with the background color, position, and related settings that are currently configured for your billboard.
- The Billboards view in InstallShield is where you add billboard files, configure billboard-related settings, and preview billboards.

Previously, only InstallScript and InstallScript MSI projects included support for billboards.

Expanded Billboard Support in InstallScript and InstallScript MSI Projects

InstallScript and InstallScript MSI projects now let you use .gif, .jpg, and .jpeg files as billboards. Previously, only .bmp files were supported. Use the Support Files/Billboards view to add billboards to your project.

Support for HTML Controls on Dialogs During InstallScript and InstallScript MSI Installations

InstallShield includes support for HTML controls on dialogs in InstallScript and InstallScript MSI projects. HTML controls enable you to use HTML markup for dialog controls.

You can include on dialogs links to Web pages, installed HTML files, and HTML support files. If an end user clicks the hyperlink on the run-time dialog, you can have the HTML page open in an Internet browser, or you can trigger other behavior that you have defined through your InstallScript code.

The HTML control lets you use any valid HTML markup, including styles to control their appearance.

Support for Specifying Action Progress Messages

To keep end users informed, installations commonly display text on the progress dialog to describe the installation's current activity. This usually accompanies the progress bar as a means of installation status. As each standard action and custom action is encountered, a message about the action is displayed on the progress dialog. This may be especially useful for actions that take a long time to execute. The same action text is also written to the installation's log file if one is created.

The Custom Actions and Sequences view has a new Action Text area that lets you specify action descriptions and details.

Setup.exe Manifests Now Have Compatibility Section to Avoid Triggering Program Compatibility Assistant on Windows 7 and Windows Server 2008 R2 Systems

If you configure your InstallShield project to create a setup launcher for your installation, the manifest that InstallShield creates for the setup launcher now includes a compatibility section. Previously, without this compatibility section, a Program Compatibility Assistant (PCA) dialog box may have appeared at the end of an installation on Windows 7 and Windows Server 2008 R2 systems. The PCA dialog box indicated that the program might not have installed correctly. This dialog box was displayed if the installation did not create the application uninstallation key. This may happen if the end user cancels the installation or the installation fails to complete successfully.

Updated Visual Studio Merge Modules

InstallShield includes updated versions of the Visual Studio 2008 SP1, Visual Studio 2005 SP1, and Visual Studio .NET 2003 SP1 merge modules that are installed in *InstallShield Program Files Folder\Modules\i386* by default. The updates include changes that were made for [Microsoft Security Bulletin MS09-035](#) (Vulnerabilities in Visual Studio Active Template Library Could Allow Remote Code Execution (969706)).

Windows Installer 5 Support for Configuring New Customization Options for Windows Services, Plus Other Service-Related Improvements

InstallShield now includes support for configuring extended customization options for Windows services. The customization options include a new delayed auto-start capability to help system startup performance, improved failure detection and recovery options to improve system reliability, and more. Windows Installer 5 supports these new options; earlier versions of Windows Installer ignore them.

To configure the new service-related settings, use the new Services node in the Advanced Settings area for a component in the Setup Design view (in installation projects) or the Components view. All of the services for a component are now grouped and listed by service name under the Services node. In addition, the previously available service-related settings are now consolidated in the same grid with the new settings. In earlier versions of InstallShield, these previously available service-related settings were split among separate subviews for the Install NT Services and Control NT Services nodes. The consolidated grid of settings makes it easier to create a component that installs, configures, starts, stops, or deletes a new or existing service during installation or uninstallation.

Validation for the Windows 7 Logo Program

InstallShield includes two new validation suites: InstallShield Validation Suite for Windows 7 and InstallShield Merge Module Validation Suite for Windows 7. These validation suites can help you verify that your Windows Installer-based installation or merge module meets the installation requirements of the Compatible with Windows 7 logo program. If you want to be able to use the Windows 7 logo artwork, your product's installation must meet the program's requirements.

If you want to configure InstallShield to perform validation with these validation suites each time that a release is successfully built: On the Tools menu, click Options. On the Validation tab, select the appropriate check boxes.

If you want to perform validation separately from the build process: On the Build menu, point to Validation, and then click the appropriate new suite.

Windows Installer 5 Support for Setting Windows Shell Properties for a Shortcut

The Shortcuts view in InstallShield now lets you specify one or more shortcut properties that need to be set by the Windows Shell at run time. For example, if you do not want the Start menu entry for one of your shortcuts to be highlighted as newly installed after end users install your product, you can set a property for that shortcut. You might want to use this property with shortcuts that are for tools and secondary products.

Windows Installer 5 includes support for setting Shell shortcut properties. Earlier versions of Windows Installer ignore those properties.

Ability to Specify a Custom .png Shortcut Icon for the Start Screen on Windows Mobile 6.5 Professional-Powered Devices

If you are creating a mobile device installation for Windows Mobile-powered devices and you plan to distribute your product through Windows Marketplace for Mobile, you need to use a 90×90 pixel Start screen icon (.png file) for your product. Both the Windows Mobile Wizard and the Smart Device Setup Wizard in InstallShield enable you to select the .png file that you want to use. When the shortcut is installed and displayed on Windows Mobile 6.5 Professional-powered devices, the Windows Mobile Shell scales down the shortcut's icon as necessary, depending on the DPI of the device.

The Shortcut Properties dialog box has a new **Use a Start screen icon (.png)** check box. If you want to use a .png file, select this check box, and then click the browse button to specify the file. The Shortcut Properties dialog box opens when you are using the Windows Mobile Wizard or the Smart Device Setup Wizard and you add a new shortcut or change the properties of an existing shortcut on the Shortcuts panel.

Windows Installer 5 Support for the MsiPrint and MsiLaunchApp Control Events

When you are adding a control to a dialog in your project, you can select one of the new events that is now available for Windows Installer 5:

- **MsiPrint**—You can add this event to a push button control that is on a dialog that has a scrollable text control. When an end user clicks the push button control, the contents of the scrollable text control are printed.
- **MsiLaunchApp**—You can add this event to a check box control on a dialog, and select the file that you want to be launched in the event's Argument setting. The check box enables end users to choose whether to run the file at the end of the installation. This event is typically used with a check box control that is on the **SetupCompleteSuccess** dialog. The check box control should include a condition that prevents the control from being displayed during uninstallation.

Windows Installer 5 includes support for these control events. Earlier versions of Windows Installer ignore these events. Therefore, if your installation is run on a system that has Windows Installer 4.5 or earlier and you want to use one or both of these new events, add a condition to the dialog controls so that they are not displayed on systems that have Windows Installer 4.5 or earlier.

Windows Installer 4.5 Prerequisites for Windows Vista Systems

The conditions in the following two InstallShield prerequisites are now configured correctly: Windows Installer 4.5 for Windows Vista (x86) and Windows Installer 4.5 for Windows Vista (x64). If one of these InstallShield prerequisites is installed on the target system during your installation, the installation no longer attempts to install it again.

Dependency Scanning and COM Extraction Now Exclude Protected Windows 7 Files and Registry Entries

The Filters.xml file that is installed with InstallShield has been updated with exclusions for files and registry keys that are protected by Windows Resource Protection (WRP) on Windows 7 systems. Therefore, if you extract COM data from COM servers, InstallShield does not extract any registry keys that are protected by WRP for Windows 7. In addition, if you run the dependency scanners in InstallShield, they do not list as dependencies any files that are protected by WRP for Windows 7.

Virtualization Pack Enhancements

The Virtualization Pack Assistants include the following enhancements:

- [Support for Creating App-V Package Upgrades](#)
- [Improved Batch Conversion Technology](#)
- [External Shortcut Support](#)
- [Windows Installer Wrapper Portability](#)
- [Support for Compressing App-V Packages](#)
- [Link to Build Output Folder Provided in Output Window](#)
- [Ability to Specify Global Setting to Identify Build Location](#)
- [Microsoft App-V Assistant OS Requirements List Now Includes Windows 7 and Windows Server 2008 R2](#)
- [Microsoft App-V Assistant Now Uses UTF-8 Encoding](#)
- [App-V Font Support](#)
- [App-V 64 KB Block Size](#)

Support for Creating App-V Package Upgrades

InstallShield now has support for creating App-V package upgrades. The Package Information page in the Microsoft App-V Assistant has a new **Upgrade Settings** link. Click this link to specify whether you want to create an upgrade. If you specify that you do want to create an upgrade, you can specify additional information about the upgrade, such as whether to append the version number to the App-V package file name. By default, App-V package upgrades are not created.

Improved Batch Conversion Technology

AdminStudio's App-V batch conversion technology has been enhanced to include additional Windows Installer configurations, resulting in more Windows Installer packages being successfully converted to App-V packages.

External Shortcut Support

AdminStudio now supports external shortcuts in App-V packages, which means that the target executable no longer needs to be included in the package. External shortcuts are especially useful for plug-ins.

Windows Installer Wrapper Portability

A Windows Installer package can be generated as part of the App-V build output. This "wrapper MSI" can be used to deploy the App-V package onto the App-V client. The .sft file can be included inside the wrapper or can be located by the App-V client based on the information in the .osd and manifest files. In AdminStudio 9.5, the wrapper MSI can now search for the .sft file in the same directory as the wrapper MSI, and, if found, it will use this location for the .sft file when deploying the App-V package to the client. This makes the wrapper MSI more portable, making it easy to move the wrapper MSI from one network location to another.

Support for Compressing App-V Packages

The Build Options page in the Microsoft App-V Assistant has a new setting that lets you specify whether you want to use compression for the data files in the App-V package. If you select Yes for this setting, InstallShield uses zlib compression for your App-V package.

Link to Build Output Folder Provided in Output Window

When you build an App-V package, the Output window now displays a link to the build output folder for the virtual package.

Ability to Specify Global Setting to Identify Build Location

InstallShield includes a new machine-wide setting that lets you specify an existing writable location where all virtual packages should be built. You may want to use this global setting if you are converting .msi packages that are in a read-only location to virtual packages.

To configure the machine-wide setting, find the `Settings.xml` file that is installed with InstallShield. In that `Settings.xml` file, enter the path as the text content for the `<GlobalBuildRedirectFolder>` element, which is a child element of the `<Virtualization>` element.

Microsoft App-V Assistant OS Requirements List Now Includes Windows 7 and Windows Server 2008 R2

The Package Information page of the Microsoft App-V Assistant now includes options for targeting Windows 7 and Windows Server 2008 R2.

Microsoft App-V Assistant Now Uses UTF-8 Encoding

InstallShield now creates the .osd, manifest, and .sprj files with UTF-8 encoding when building App-V packages; this matches the behavior of the Microsoft Application Virtualization Sequencer. Previously, InstallShield created these files with UTF-16 encoding, which seemed to lead to parser errors on some systems that had Cumulative Update 1 for Microsoft Application Virtualization 4.5.

App-V Font Support

Custom fonts included in installation packages will now be supported for App-V conversions. Therefore, the fonts will be available when running the applications (.osds) in the generated App-V packages.

App-V 64 KB Block Size

The data block size has been increased from 32K to 64K. This is consistent with Microsoft Sequencer 4.6. One of the advantages of this change is that App-V packages will feature slightly better compression.

Repackager Enhancements

AdminStudio Repackager 9.5 includes the following enhancements:

- [Build Engine Upgrade](#)
- [Updated Repackager Exclusions List for Windows 7](#)
- [Updated Repackager Installation Monitoring Method Exclusions for Windows 7](#)

- [Enhanced Support for Capturing Side-by-Side Assemblies](#)
- [Enhanced User Folder Handling](#)
- [Enhanced International Language Support](#)

Build Engine Upgrade

Repackager now uses the Build Engine from InstallShield 2010 SP1.

Updated Repackager Exclusions List for Windows 7

The default Repackager exclusions list has been updated for Windows 7. If you capture an application on a Windows 7 operating system, Repackager will automatically exclude all OS files and registry entries.

Updated Repackager Installation Monitoring Method Exclusions for Windows 7

The process exclusions used by the Repackager Installation Monitoring method has been updated to include the process exclusions for Windows Vista and Windows 7.

Enhanced Support for Capturing Side-by-Side Assemblies

Repackager now provides improved functionality when capturing side-by-side assemblies.

Enhanced User Folder Handling

In Windows 7, user data folders have been moved. Therefore, in AdminStudio 9.5, the handling of Repackager user data has been improved to accommodate these changes.

Enhanced International Language Support

AdminStudio provides enhanced international language support:

- Enhanced repackaging on international platforms.
- Improved repackaging of international applications that use double-byte character sets.
- Management of international applications in the Application Catalog.

AdminStudio Editions

AdminStudio enables systems administrators to rapidly prepare error-free applications that deploy seamlessly into any enterprise environment through a structured process built on application management best practices.

To meet the needs of organization of every size, AdminStudio 9.5 is available in the following editions.

Table 1-3 • AdminStudio Editions


Edition	Description
Standard	<p>Standard Edition provides basic repackaging and editing capabilities. Standard Edition includes:</p> <ul style="list-style-type: none"> • Repackager • InstallShield Editor • Application Isolation Wizard • Tuner • FLEXwrap • Distribution Wizard
Professional	<p>The Professional Edition is designed for professional packagers. It includes application management, testing, and conflict resolution tools, as well as additional distribution options.</p> <p>Professional Edition includes all of the Standard Edition tools plus:</p> <ul style="list-style-type: none"> • Application Manager • ConflictSolver • QualityMonitor • Predeployment Test • OS Snapshot Wizard • Configuration Manager Web Console <p>Documentation on features that are only available in AdminStudio Professional and Enterprise Editions includes the following note:</p> <div data-bbox="581 1297 620 1339">  </div> <hr/> <p>Edition • <i>This feature is included with AdminStudio Professional and Enterprise Editions.</i></p>

Table 1-3 • AdminStudio Editions (cont.)

Edition	Description
Enterprise	<p>The Enterprise Edition is the enterprise-class solution. It includes Web-based server access with additional security, reporting, and automation tools. Enterprise Edition includes all Professional Edition tools plus:</p> <ul style="list-style-type: none"> • PackageExpert • OS Security Patch Wizard • Security Console • Job Manager • Report Center • Software Repository <p>Documentation on features that are only available in AdminStudio Enterprise Edition includes the following note:</p> <div data-bbox="581 821 621 867"></div> <hr/> <p>Edition • This feature is included with AdminStudio Enterprise Edition.</p>
Virtualization Pack	<p>AdminStudio offers a Virtualization Pack that includes Microsoft App-V, VMware ThinApp, and Citrix XenApp virtualization support. The following additional components are available:</p> <ul style="list-style-type: none"> • InstallShield Microsoft App-V Assistant • InstallShield ThinApp Assistant • InstallShield Citrix Assistant • Repackager Windows Installer Converter <p>Documentation on these features includes the following note:</p> <div data-bbox="581 1297 621 1344"></div> <hr/> <p>Edition • This feature is included in the AdminStudio Virtualization Pack.</p> <p>Also, documentation on the Batch Convert option of Windows Installer Converter includes the following note:</p> <div data-bbox="581 1493 621 1539"></div> <hr/> <p>Edition • The Batch Convert option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.</p>



Note • For a comprehensive list of the features available in the AdminStudio Standard, Professional, and Enterprise Editions, plus the add-on features available for Enterprise Edition, see the AdminStudio Web site.

Activating AdminStudio

When you launch AdminStudio or one of its tools for the first time, you are notified that you are using a time-limited trial version, and you are given the opportunity to evaluate the product or to activate it by entering a valid Serial Number for an AdminStudio Edition.



Task: *To activate AdminStudio:*

1. Install AdminStudio, as described in the *AdminStudio Client Tools Installation Guide*.
2. Launch AdminStudio or one of its tools. A dialog box opens, stating that you are using a time-limited trial version.
3. Select **Activate or Purchase AdminStudio** and click **Next**. The **AdminStudio Product Activation** dialog box opens, prompting you to enter a serial number.
4. Enter the serial number of the edition you purchased and click the **Activate** button. After a few seconds, you will receive a message that activation was successful.
5. Click **Finish**. AdminStudio will launch.



Note • If you want to evaluate the AdminStudio Enterprise Server Web tools (Job Manager, Report Center, Security Console), you need to obtain an Enterprise Server Evaluation Serial Number from an AdminStudio Sales Representative.

Ports Used in Activation

AdminStudio product activation uses ports 80 (HTTP), 443 (HTTPS), and 8443. If these ports are locked down or if you do not have an available internet connection, you can activate AdminStudio using one of the following alternative methods:

- **Self-hosted activation using FLEXnet License Server**—Your organization can choose to purchase and use a self-hosted FLEXnet License Server to manage their AdminStudio licenses. Using the FLEXnet License Server eliminates the need to manually activate each installation of AdminStudio over the internet.
- **Offline activation**—You can perform offline activation using email.

Silent Activation

AdminStudio supports silent activation, enabling you to use public Windows Installer properties to enter the Serial Number automatically and activate during installation. To enable silent activation, you need to set the PRODUCTID and ASACTSTATUS public Windows Installer properties. The following is an example of how to accomplish this via the command line:

```
<AdminStudio_Install_Directory>Setup.exe /v"PRODUCTID=XXXXXXX-XXX-XXXXXXXXXX ASACTSTATUS=1"
```

You can additionally make the installation completely silent by adding the /s and /qn switches as follows:

```
<AdminStudio_Install_Directory>Setup.exe /s /v"PRODUCTID=XXXXXXX-XXX-XXXXXXXXXX ASACTSTATUS=1 /qn"
```



Note • Consult the Windows Installer Help Library for any additional Windows Installer command line options.

Evaluating AdminStudio

You can choose to evaluate AdminStudio for 21 days. By clicking **Continue to Evaluate AdminStudio** dialog box that opens when you launch AdminStudio, you can begin evaluating the AdminStudio 9.5 Enterprise Edition client tools. To evaluate the AdminStudio Enterprise Server Web tools, you need to contact AdminStudio Sales and obtain an evaluation serial number.

Information about evaluating AdminStudio is presented in the following sections:

- [Evaluating AdminStudio Enterprise Edition Client Tools](#)
 - [Evaluating AdminStudio's Microsoft App-V Support](#)
 - [Evaluating the Windows Installer Converter Batch Convert Option](#)
- [Evaluating the AdminStudio Enterprise Server Web Tools](#)

Evaluating AdminStudio Enterprise Edition Client Tools

To evaluate the Enterprise Edition client tools, perform the following steps.



Task:

To evaluate the AdminStudio Enterprise Edition client tools:

1. Install AdminStudio, as described in the *AdminStudio Client Tools Installation Guide*.
2. Launch AdminStudio. A dialog box opens, stating that you are using a time-limited trial version.
3. If you want to evaluate AdminStudio, select **Continue to Evaluate AdminStudio** and click **Next** (or just wait ten seconds). The product will launch.
 - When you run AdminStudio in this trial mode, all of its features are fully available.
 - Each time you open AdminStudio while you are in evaluation mode, this dialog box shows you how many days are left in your trial period.
4. If you have five or fewer days left in your trial period, the dialog box remains open, requiring you to click before you can proceed. Do one of the following:
 - a. If your trial period is not over, you can continue to use AdminStudio by selecting the **Continue to Evaluate AdminStudio** option and clicking **Next**.
 - b. If you have already purchased a serial number or want to purchase one online, select **Activate or Purchase AdminStudio** and click **Next**.

Evaluating AdminStudio's Microsoft App-V Support

While evaluating the AdminStudio Enterprise Edition client tools, you will be able to convert a Windows Installer package to an App-V application using both the Windows Installer Converter and the InstallShield App-V Assistant. However, an App-V application built using an evaluation version of AdminStudio will display the following message every time it is launched:

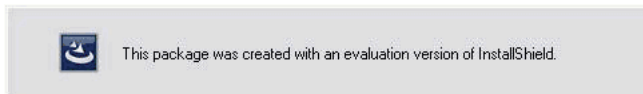


Figure 1-1: Evaluation Version Message

After purchasing the AdminStudio Virtualization Pack, you will be able to remove this message by rebuilding the App-V application.

Evaluating the Windows Installer Converter Batch Convert Option

The **Batch Convert** option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.

When using an evaluation version of AdminStudio, you will be able to use the **Batch Convert** option to convert a directory full of Windows Installer packages into individual virtual packages, but the batch conversion will be limited to ten packages; only the first ten Windows Installer packages that Windows Installer Converter encounters will be converted to virtual applications.

Evaluating the AdminStudio Enterprise Server Web Tools

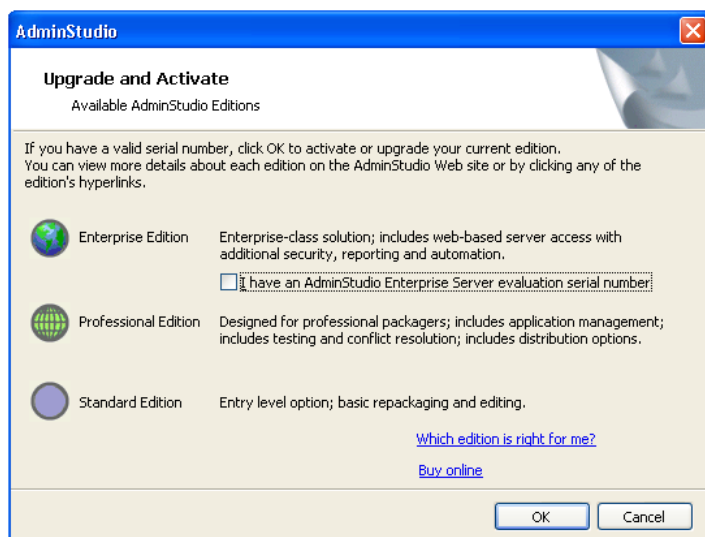
If you want to evaluate the AdminStudio Enterprise Server Web tools (Job Manager, Report Center, Security Console), you need to obtain an AdminStudio Enterprise Server Evaluation Serial Number from an AdminStudio Sales Representative.



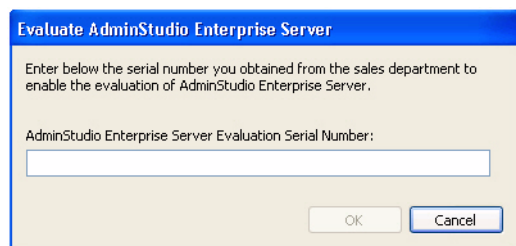
Task:

To evaluate the AdminStudio Enterprise Server Web tools:

1. Follow the steps in [Evaluating AdminStudio Enterprise Edition Client Tools](#).
2. Contact an AdminStudio Sales Representative and obtain an AdminStudio Enterprise Server Evaluation Serial Number.
3. Open the AdminStudio interface and select **About AdminStudio** from the **Help** menu.
4. Click **Activate**. The **Upgrade and Activate** dialog box opens.



5. Select the **I have an AdminStudio Enterprise Server evaluation serial number** option and click **OK**. The **Evaluate Enterprise Edition** dialog box opens.



6. Enter the serial number and click **OK**. You will be prompted to install the AdminStudio Enterprise Server on the current machine.
7. Click **Yes**. The AdminStudio Enterprise Server installation begins.
8. Install AdminStudio Enterprise Server, as described in the *AdminStudio and Workflow Manager Installation Guide*, which is available on the AdminStudio Downloads site.

Upgrading Your Product Edition

An upgrade feature has been built-in to AdminStudio that allows you to activate features in a higher Edition without re-installing the application. You just need to enter a Serial Number for the upgrade that you purchased, and the features of that Edition are immediately unlocked and are available to you.

To upgrade, perform the following steps:



Task: *To upgrade your AdminStudio Edition:*

1. Contact an AdminStudio Sales Representative and purchase a Serial Number for the desired Edition.
2. Launch AdminStudio.
3. On the **Help** menu, click **About AdminStudio**. The **About AdminStudio** dialog box opens.
4. Click the **Upgrade** button. The **Upgrade and Activate** dialog box opens.
5. Click **OK** to upgrade your edition. The **AdminStudio Product Activation** dialog box opens, prompting you to enter the serial number of the edition that you want to upgrade to.
6. Enter the serial number of the edition you purchased and click the **Activate** button. After a few seconds, you will receive a message that activation was successful. The functionality of the upgraded edition is immediately available to you.

Using Help

Help is available both from the AdminStudio interface **Help** menu and directly from certain individual interface elements.

When you have questions about this product, first consult the AdminStudio Help Library, which is the complete user's guide for using AdminStudio.

Web-Based Online Help

Web-based online help is available to you 24 hours a day, seven days a week, on our Web site at:

<http://helpnet.flexerasoftware.com>

Documentation as PDF

AdminStudio documentation is also available as a PDF. Visit the AdminStudio Downloads site at:

<http://www.flexerasoftware.com/downloads.htm>

Help Conventions








In this documentation, reader alert and style conventions are used to bring your attention to specific information or help you identify information.

- [Reader Alert Conventions](#)
- [Style Conventions](#)

Reader Alert Conventions

Reader alerts are used throughout this documentation to notify you of both supplementary and essential information. The following table explains the meaning of each alert.

Table 1-4 • Reader Alert Conventions

Image	Alert Name	Description
	Note	Notes are used to draw attention to pieces of information that should stand out.
	Important Note	Important notes are used for information that is essential for users to read.
	Caution	Cautions indicate that this information is critical to the success of the desired feature or product functionality.
	Tip	Tips are used to indicate helpful information that could assist you in better utilizing the desired function or feature.
	Best Practices	Best Practices alerts instruct you on the best way to accomplish a task.
	Edition-Specific Note	Edition-specific notes indicate that the information applies to a specific edition of a product (such as Professional or Premier edition).
	Procedure	The Procedure graphic indicates that procedural instructions follow.

Style Conventions

The following style conventions are used throughout this documentation.

Table 1-5 • Style Conventions

Style	Example	Description
User Interface Elements	On the File menu, click Open .	User interface elements appear in bold when referenced in tasks.
Variables	<i>fileName</i>	Variables appear in italics.
Code	<code>#define HWND_BROADCAST 0xffff</code>	Code snippets appear in a monospace typeface.

Table 1-5 • Style Conventions (cont.)

Style	Example	Description
User-Inputted Text	Type <code>\$D(install)</code> .	Text that is to be entered as a literal value is displayed in a monospace typeface, in bold, and in blue.
File Names and Directory Paths	My files are located in the C:\MyDocuments\SampleCode directory.	File names and directory paths are presented in a monospace typeface.
.INI File Text	Insert the line LimitedUI=Y into the file to display only the Welcome dialog box when the Windows Installer package is run.	Text in .INI files is presented in a monospace typeface.
Command-Line Statements	To run the installation silently, enter: Setup.exe /s /v/qn	Command-line statements and parameters are presented in a monospace typeface.
Environment Variables	Set the value of the windir environment variable.	Environment variables are presented in a monospace typeface.
Examples	Create two groups, one called Admin and the other called General .	Examples are presented in bold.
Functions	FeatureAddItem adds a new feature to a script-created feature set.	Functions are presented in bold.
Properties	In the Name property, enter a name for this custom control that is unique among all of the controls in your project.	Properties are presented in bold.
Screen Output	If you type an incorrect parameter, the message <code>The system cannot find the path specified.</code> is displayed.	Screen output (from a log file or from the console) is displayed in a monospace typeface, and in blue.
Links	Obtain the latest white papers, project samples, and more from: http://www.mycompany.com/downloads.htm	Links appear in blue.

Contacting Us

You may contact us from anywhere in the world by visiting our Web site at:

<http://www.flexerasoftware.com>

Chapter 1: AdminStudio 9.5 Help Library

Contacting Us

Using the Process Assistant

The Process Assistant is a procedure-based alternative interface to the standard AdminStudio interface that guides you through the process of preparing a new application for enterprise use. Using the Process Assistant, you can perform the following tasks:

- Packaging and feature customization
- Conflict resolution
- QA testing
- Deployment hand-off

The [ConflictSolver Process Assistant](#), which is integrated into the main Process Assistant but which can also be opened from the ConflictSolver interface, guides you through the process of identifying and resolving conflicts between Windows Installer packages.

Using the Process Assistant decreases the time required for new AdminStudio users to prepare software packages for deployment, and also enables veteran systems administrators to efficiently jump start a new packaging project. It provides assistance on what steps to perform next, based on your previous selections.

Process Assistant Pages

The Process Assistant Pages take you through the steps typically performed when preparing an application for deployment.

Table 2-1 • Process Assistant Pages







Task	Page Name	Description
	Home Page	Select an existing non-MSI or legacy application to convert to Windows Installer format and launch the repackaging process.
	Package Page	Select an existing non-MSI or legacy application to convert to Windows Installer format and launch the repackaging process.

Table 2-1 • Process Assistant Pages (cont.)

Task	Page Name	Description
	Customize Page	Customize your installation by directly modifying the Windows Installer package or by creating a transform.
	Resolve Conflicts Page	Select the Windows Installer package you want to check for conflicts against packages already in the Application Catalog, and launch the ConflictSolver Process Assistant .
	Test Page	Test the package installation on the targeted deployment environment to ensure that it works as expected.
	Distribute Page	Prepare your package for distribution.

Opening the Process Assistant

To open the Process Assistant, click on the **Assistant** tab in the AdminStudio interface. You can also open it by clicking the Process Assistant link on the **Start Page**.

Also, you can open the ConflictSolver Process Assistant from the ConflictSolver interface by selecting **Process Assistant** on the **View** menu or by clicking the Process Assistant toolbar button:



Using the Process Assistant

The Process Assistant is a procedure-based alternative interface to the standard AdminStudio interface. To open the Process Assistant, click the **Assistant** tab in the AdminStudio interface. The **Process Assistant Welcome Page** includes a packaging diagram to help you visualize the steps involved in preparing a package for deployment.

Navigating in the Process Assistant

You move through the Process Assistant using the navigation bar at the bottom of the page:

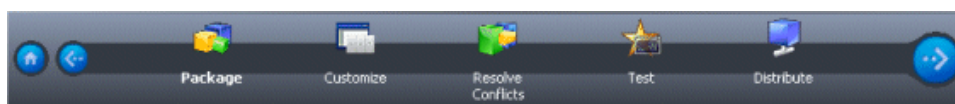


Figure 2-1: Process Assistant Navigation Bar

To navigate from one task page of the Process Assistant to another, click on buttons in the navigation bar:

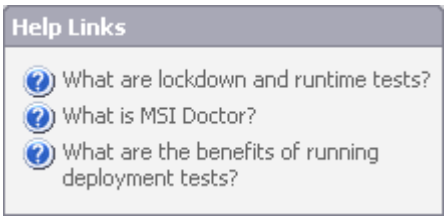

- To navigate directly to a specific page, click the appropriate icon in the navigation bar at the bottom of the page.

- To follow the Process Assistant steps sequentially, click the **Next** or **Back** arrow buttons to move forward or backward.
- To move back to the Home Page and view the diagram, click the Home button on the navigation bar.

List Controls

Each page in the Process Assistant contains a **Help Links** list control and, when applicable, a **More Options** list control to assist you:

Table 2-2 • Help Links and More Options List Controls

	<p>Help Links</p> <p>This list provides links to help topics pertinent to the current Process Assistant page.</p>
	<p>More Options</p> <p>Provides additional configuration options relating to the specific area on the Process Assistant page. These are less common options that complete the functionality of the Process Assistant.</p>

Opening the AdminStudio Interface

To open the AdminStudio interface and Tools Gallery (and hide the Process Assistant), click on the **Tools** tab.

Home Page

The **Home Page** displays a diagram outlining the major tasks that are usually performed when a package is being prepared for deployment. On this Page, you can also select an installation program, and when you click **Start**, the Process Assistant will open the Page that is most appropriate for the file selected. For example, if you select:

- **Windows Installer package (*.msi)**—The **Customize Page** opens, bypassing the **Package Page**, and you are prompted to modify this package either directly or by creating a transform. Because this package is already in Windows Installer format, the repackaging step would not be necessary.
- **Setup program file (*.exe)**—The **Package Page** opens and you are prompted to repackage this setup program file into Windows Installer format.

Package Page

On the **Package Page**, you select an existing non-MSI or legacy application to convert to Windows Installer format and launch the repackaging process.



Task: *To select a package for repackaging:*

1. Click on the **Package** icon in the navigation bar. The **Package Page** opens.
2. Click **Browse** and select the package or application that you want to convert. Additional packaging fields appear on the **Package Page**.
3. In the **Product Information** area, the **Product name** and **Version number** are automatically populated based upon information in the selected file. Make any desired edits.
4. Enter any [Repackaging Wizard Command-Line Options](#) that you wish to include.
5. Select one of the [Repackaging Methods](#): **Install Monitoring** or **Snapshot**.
6. Click **Browse** and select a Repackaging Wizard output directory, the directory where you want the Repackaging Wizard to save the converted Windows Installer (.msi) package.
7. Click **Repackage** to launch the [Repackaging Wizard](#)

Customize Page

On the **Customize Page**, you can customize your installation by directly modifying the Windows Installer package or by creating a transform, a file that you append to the package which will customize the file at the time of installation.



Task: *To customize an installation:*

1. Click on the **Customize** icon in the navigation bar. The **Customize Page** opens.
2. Click **Browse** and select the Windows Installer package that you want to customize.
3. Under **Customization Options**, select to either modify the original Windows Installer file or to create a transform.
4. If you selected the **Create a transform file** option, enter the name of the transform file you are creating or click **Browse** to select a transform file that you are replacing.
5. Click **Customize** to begin the customization process. InstallShield Editor is launched either in Direct MST Mode (to create a transform) or Direct Edit Mode (to modify the original .msi file).
6. Customize the package using InstallShield Editor.
7. After customization is complete, click **Import** on the **Customize Page** to import the converted package into the Application Catalog.

Adding Transforms to an MSI Package

When you prepare a package for distribution, you need to package the transforms with the base .msi package. You can perform this task using InstallShield Editor or Tuner. See [Customizing and Authoring Installations Using InstallShield](#) or [Preparing Packages for Distribution Using Distribution Wizard](#).

You can also specify a transform at the command line:

```
msiexec /i "yourmsi.msi" TRANSFORMS="yourtrans.mst"
```

Resolve Conflicts Page







Edition • ConflictSolver is included with AdminStudio Professional and Enterprise Editions.

Although a Windows Installer package or merge module may be built to guidelines put forth by Microsoft, it is possible that the interaction between packages, or between a package and the base operating system, may cause unwanted results in your production environment. You perform conflict detection to identify these conflicts before you deploy packages, and resolve the problems before they affect your end users.

On the Resolve Conflicts page, you select the Windows Installer package you want to check for conflicts against packages already in the Application Catalog.



Task: *To select a package for conflict detection:*

1. Click on the **Resolve Conflicts** icon in the navigation bar. The **Resolve Conflicts Page** opens.
2. Click **Browse** to locate and select the package.
3. Should that package require transforms, click the **New** button () to launch the **Select Transform Files** dialog box and select a transform file.
 - If the package requires multiple transforms, you can repeat the procedure as necessary.
 - The order in which transforms are applied can be changed by selecting a transform and clicking the **Move Up** () and **Move Down** () buttons.
 - Delete a transform file from the **Transforms** list by selecting the transform and clicking the **Delete** button ().
4. From the ConflictSolver **Operation** list, select **Detect conflicts for this package** or **Resolve conflicts for this package**.
5. Click **Perform** to launch conflict detection for this package.

Importing Applications into the Application Catalog

You can import Windows Installer packages (.msi) with any associated transforms (.mst) or patches (.msp) and merge modules (.msm) into the Application Catalog. Additionally, you can import OS snapshots (.osc), Marimba Native Channel Packager (.ncp) files and other non-MSI based setup programs.

Once you import these files, they become available for conflict identification. Although you can identify conflicts between an external package and packages already in ConflictSolver, it is strongly recommended you import the package first and perform conflict identification after import. Conflict information is only persisted for the last conflict identification performed between an external package and packages in ConflictSolver, and performance is better when performing conflict identification between internal packages.

The Application Catalog also serves as a repository for all applications used in your organization. You can create custom installations of a common Windows Installer package to distribute to different departments in your organization, each installation including certain features that are appropriate for the department and excluding certain features that are not appropriate.

Software Repository

If you are connected to an Application Catalog that has the Software Repository enabled, and you want to add the package that you are importing to the Software Repository, select the **Add package to the Software Repository** option on the **Organize Package Page** of the ConflictSolver Process Assistant. See [About the Software Repository](#) for more information.

Application Conflicts

Although a Windows Installer package or merge module may be built to guidelines put forth by Microsoft, it is possible that the interaction between packages, or between a package and the base operating system, may cause unwanted results in your production environment. You can use ConflictSolver to identify these conflicts before you deploy packages, and resolve the problems before they affect your end users.

ConflictSolver allows you to check for a variety of conflict types, including file, component, and registry conflicts. In many cases, ConflictSolver can resolve the issues automatically. ConflictSolver has rules to detect conflicts involving: Components, Files, Registry Entries, Shortcuts, INI Files, ODBC Resources, NT Services, File Extensions, and Product Properties.

Automatic and Manual Conflict Resolution

Automatic Conflict Resolution

To identify conflicts between multiple installation packages, ConflictSolver uses Application Conflict Evaluators (ACEs). Some ACE failures can be resolved automatically by ConflictSolver. These automatic resolution techniques are defined in Conflict Application Resolution Definition (CARD) rules. ACEs that can be automatically resolved by CARs are referred to as Best Practice ACEs: ACE02, ACE04, ACE05, ACE06, ACE07, ACE15, ACE18, ACE19, ACE20. CARs resolve errors found by the corresponding ACE of the same number.

You can choose to automatically resolve ACEs on the **Conflicts** view for the product. Select the conflict you want to resolve and click Resolve. Alternatively, click Resolve All to resolve all resolvable conflicts for the selected conflict category. See [Resolving Best Practice ACE Errors](#).

Manual Conflict Resolution

Due to their complexity, some conflicts require manual resolution using Tuner or InstallShield Editor.

For information on manual solutions, consult the [Application Conflict Evaluators \(ACEs\)](#) topic or individual ACE topics. These topics can provide valuable information regarding what is causing the conflict, so you can use Tuner or InstallShield Editor to solve it.

Once you have resolved detected conflicts manually, you can re-import the package into ConflictSolver.

ConflictSolver Process Assistant



Edition • ConflictSolver is included with AdminStudio Professional and Enterprise Editions.

The ConflictSolver Process Assistant, a procedure-based alternative interface to the ConflictSolver Application Manager interface, guides you through the process of identifying and resolving conflicts between Windows Installer packages. The ConflictSolver Process Assistant is integrated into the main Process Assistant via the **Resolve Conflicts Page** task, and can also be launched separately from the Tools Gallery.

The following standard ConflictSolver tasks are included in the ConflictSolver Process Assistant:

Table 2-3 • ConflictSolver Process Assistant Tasks









Task	Description
 Select a Package Page	Select a current package or import a new one.
 Extended Attributes Page	View and edit optional attributes for packages.
 Validation Page	Validate the package using selected Internal Consistency Evaluators (ICEs).
 Best Practices Page	Check the selected package to see if it meets Windows Installer “best practices” rules, and then choose to resolve any best practice conflicts that are found.
 Detect Conflicts Page	Check the selected package against target packages to determine if conflicts would occur if all of these packages were installed on the same system.

Table 2-3 • ConflictSolver Process Assistant Tasks

Task	Description
 Resolve Conflicts Page	View all detected conflicts and choose to resolve any conflicts that have an automatic resolution.
 Reporting Page	Generate various types of package reports listing data about conflicts, validation, extended attributes, files, and registry entries.
 Organize Package Page	Relocate packages within the Application Catalog.





Select a Package Page

On the **Select a Package Page**, you select a source package for conflict detection: the Windows Installer file you want to check for conflicts against packages in the Application Catalog. You can choose either a package that is already in the Application Catalog (a **Current** package) or a package that you want to import into the Application Catalog prior to conflict detection (a **New** package). If you choose to import a package, you can also choose to import transform files and patches along with it.

Importing a New Package



Task: *To import a new package:*

1. Launch the ConflictSolver Process Assistant. The **Select a Package Page** opens.
2. Click **Browse** to locate and select the Windows Installer package.
3. Should that package require transforms, set the **Would you like to include transforms?** option to **Yes** to enable the transform fields, and then select transform files:
 - To locate and select a transform file, click the **New** button ().
 - If the package requires multiple transforms, you can repeat the procedure as necessary.
 - The order in which transforms are applied can be changed by selecting a transform and clicking the **Move Up** () and **Move Down** () buttons.
 - Delete a transform file from the **Transform Files** (.mst) list by selecting the transform and clicking the **Delete** button ().
4. Should that package require patches, set the **Would you like to include patch files?** option to **Yes** to enable the patch file fields, and then select the patch files that you want to include using the **New**, **Move Up**, **Move Down**, and **Delete** buttons, as described above.

5. Click **Import**. The package is imported into the **Draft** group of the Application Catalog and the **Validation Page** opens.

Selecting a Current Package



Task: *To select a current package:*

1. Launch the ConflictSolver Process Assistant. **The Select a Package Page** opens.
2. Select **Current**. The **Application Catalog Group** tree is displayed.
3. Expand the **Group** tree and select the package that you want to use as a source package in conflict detection.
4. Click **Continue**. The **Validation Page** opens.

Current and New Packages

On the **Select a Package Page**, you choose the package that you are going to process using the ConflictSolver Assistant. You can choose to use either a Current or New package:

- **Current**—Package that has already been imported into the Application Catalog.
- **New**—Package that will be imported into the **Draft Group** of the Application Catalog, a working Group used for packages in the process of being prepared for distribution.

After you have selected a package, you can use it in the **Validation**, **Best Practices**, **Detect Conflicts**, **Resolve Conflicts**, **Reporting**, and **Organize Package** tasks of the ConflictSolver Assistant.

Draft Packages

When you import a package into the Application Catalog using the ConflictSolver Process Assistant, that package is automatically imported into a special group called Draft, a temporary working group that isolates an imported package that has not yet been checked for conflicts from the rest of the packages in the Application Catalog. The Draft group is only visible when you select Current on the **Select a Package Page** of the Process Assistant.

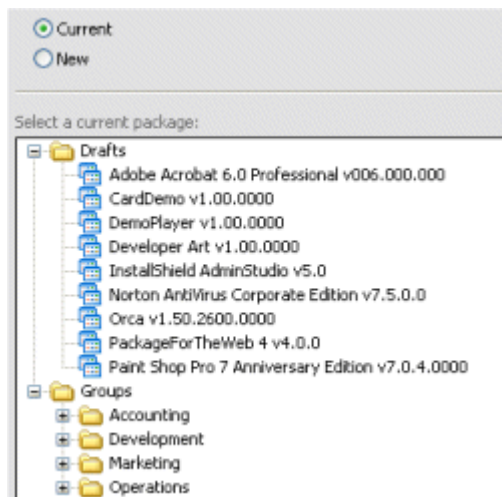


Figure 2-2: Draft Group visible on the Select a Package Page

By placing these newly imported packages into the Draft group, you are indicating that these packages are in the process of being prepared for distribution, and should not be used as target packages in conflict detection.

After you have performed validation, conflict detection, and conflict resolution on a Draft package, and it is ready for distribution at your organization, you should then move it out of the Draft group using the **Organize Package Page** of the Process Assistant. The package will then be visible on the Application Manager View.



Note • While it is possible to move a package out of the Draft group before conflict detection and resolution have been performed, it is not recommended.

Creating a New Group

You can create a new group using the **New** button on the **Organize Package Page** of the Process Assistant.



Task: *To create a new group:*

1. If a package is not currently selected, go to the **Select a Package** page and select a package.
2. Click on the **Organize Package** button in the navigation bar. The **Organize Package Page** opens.
3. Click **New**. The **New Group Properties** dialog box opens.
4. Enter a **Name**, **Description** and any desired **Comments** for the new group and click **OK**. The new Group is listed in the group tree and is automatically selected.

Configuring Import Options

Before importing a new package, you should review the following options on the **Import** tab of the ConflictSolver **Options** dialog box:

- **Import Binary Data & Import Custom Tables**—If you do not expect to run user-defined ACE rules, then you do not need to import either binary data or custom tables.
- **Check for Validation Errors & Check for Conflicts**—Checking packages for validation at the time of import is often unnecessary. Also, conflict detection at the time of package importation will become an exponentially slow operation as the size of your Application Catalog increases, since this option causes the package to be conflict checked against all existing packages in the Application Catalog.

By deselecting these import options on the **Import** tab of the ConflictSolver **Options** dialog box, import performance will improve.

Patch Files

AdminStudio supports the import of Windows Installer packages (.msi) with associated patches (.msp). A Windows Installer patch (.msp file) is a file used to deliver updates to Windows Installer applications. A patch is a self-contained package that contains all the information required to update an application.

A patch package contains the actual updates to the application and describes which versions of the application can receive the patch. A patch package does not include a database like a regular installation package (.msi file). Patches contain at minimum two database transforms. One transform updates the information in the installation database of the application. The other transform adds information that the installer uses for patching files.

Extended Attributes Page

Extended attributes are optional properties for packages that you can view and edit. Because you can manually create the description file, you have the flexibility to include information about each package that may be specific to your organization—such as the users or business groups that receive the package.

The attributes listed on the Extended Attributes match those defined in the [Extended Attribute Description File](#) that is currently selected on the **Extended Attributes** tab of the **Options** dialog box.

On the **Extended Attributes** page, you can edit any of the listed attributes for the selected package by clicking on the attribute name.



Task: *To enter or edit a package's extended attributes:*

1. If a package is not currently selected, go to the **Select a Package Page** and select a package.
2. Click on the **Extended Attributes** button in the navigation bar. The **Extended Attributes Page** opens, listing all of the attributes defined in the Extended Attributes Description file, and listing any values that have previously been entered.
3. To edit an attribute, click the attribute link. A dialog box opens and prompts you to enter text, make a selection from a list, or upload a file (depending upon how the attribute was defined in the Extended Attribute Description File).
4. Perform the desired action and click **OK**. When you return to the **Extended Attributes Page**, the changes that you made are displayed.

Validation Page

On the Validation Page, you compare the package against selected Microsoft Internal Consistency Evaluators (ICEs), custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards.



Task: *To validate a package:*

1. After selecting a package on the **Select a Package Page** of the ConflictSolver Process Assistant, the **Validation Page** opens. You can also open the **Validation Page** by clicking the Validation icon in the navigation bar.
2. Click **Validate** to begin the validation process. Results are displayed in the **Validation Results** list and on the **Validate** tab of the Output window.

If you do not want ConflictSolver to check for a particular ICE error during subsequent validations of a package, you can choose to suppress it by selecting the ICE error and selecting **Suppress** from the context menu.

3. When you suppress an ICE error, it is listed in gray on the ConflictSolver **Validation View** and is not executed in subsequent validations. You might choose to suppress an ICE error that is a known issue at your organization which does not need additional corrections.
4. You can make a selection from the View list to view only ICE Errors that are Enabled, only those that are Suppressed, or All ICE Errors.

Microsoft Internal Consistency Evaluators (ICEs)

When ConflictSolver validates a package, it compares a package against Microsoft ICEs (Internal Consistency Evaluators), custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards.

ICEs are written in VBScript, JScript, or as a DLL or EXE. When these custom actions are executed, they scan the database for entries in database records that are valid when examined individually but that may cause incorrect behavior in the context of the whole database.



Note • For a full listing and description of all Microsoft ICEs, see [ICE Reference](#) in the Microsoft MSDN Library.

Validation Error Severity Ratings

When ConflictSolver validates a package, it compares a package against Microsoft ICEs (Internal Consistency Evaluators), custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards. ICE errors have the following degrees of severity:

- **Errors**—Database authoring that causes incorrect behavior. An example of an error would be duplicate component GUIDs, which would cause the Installer to incorrectly register components.
- **Warnings**—Database authoring that can cause incorrect behavior. Warnings can also report unexpected side-effects of database authoring. An example of a warning would be entering the same property name in two conditions that differ only by the case of letters in the name. Because the Installer is case-sensitive, the Installer treats these as different properties.
- **Informational**—Information from the ICE that do not indicate a problem with the database. Often they are information about the ICE itself, such as a brief description. They can also provide progress information as the ICE runs.

Configuring Validation Options

Validation options are set on the **Validate** tab of the ConflictSolver **Options** dialog box. To fine tune validation for your organization, you may want to adjust some of the following settings.

Excluding Specific ICEs from Execution During Validation

You can permanently exclude the execution of specific ICEs that are listed in the selected .cub file.



Task: *To exclude the execution of specific ICEs during validation:*

1. In ConflictSolver, select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Select the **Validate** tab from the dialog box. The **Validate** tab opens.
3. In the MSI Input files and MSM Input files areas, list the ICEs that you want to exclude in the ICE Rules field, separated by semicolons (;), such as:

ICE07; ICE09; ICE22


All of the ICEs in the specified .cub file will be executed during validation except for the ICEs listed here.

Changing the Default Validation File

By default, ConflictSolver compares packages against the full set of Microsoft ICEs. In the overwhelming majority of cases, this is the file you will want ConflictSolver to use when performing validation. However, there may be times you want to compare your base package against a different validation file, depending on your needs.



Task: *To specify a different validation file to use:*

1. Select Options from the Tools menu. The **Options** dialog box opens.
2. Select the Validate tab from the dialog box. The **Validate** tab opens.
3. In the MSI Input files and MSM Input files areas, click the Browse button () next to the Cub File field to locate the validation file (.cub) that you would like to use.
4. To permanently exclude the execution of specific ICEs, list them in the ICE Rules field, separated by semicolons (;). Otherwise, all rules will be used. See [Excluding Specific ICEs from Execution During Validation](#) for more information.

Suppressing Warnings and Informational Messages

To suppress Warnings and Informational Messages from being displayed in the Validate tab of the **Output** window during validation, you can clear the Warnings and Info check boxes on the **Validate** tab of the ConflictSolver Options dialog box. It is strongly recommended you do not suppress warning messages so you can ensure proper validation of packages before importing them into ConflictSolver.

Best Practices Page

On the Best Practices Page, you check a package to see if it meets Windows Installer “best practices” rules, and then you are given the opportunity to resolve any conflicts that are found. See [Best Practice ACEs](#) for more information.



Task: *To check a package against Best Practice ACEs:*

1. In the ConflictSolver Process Assistant, click the **Best Practices** icon in the navigation bar. The **Best Practices Page** opens.
2. Click **Detect** to check the selected package. Any conflicts that are found are listed in the Best Practice ACE list, and are also listed in the **Conflicts** tab of the Output Window.
3. Select the conflicts that you would like to resolve and click **Resolve**.

Best Practice ACEs

A best practice ACE is an ACE that operates like an ICE to enforce Microsoft best practices. While an ICE is a custom action written by Microsoft which is used during validation to determine if an installation package is built according to Windows Installer standards, an ACE is a custom action included with ConflictSolver that is also used to enforce Windows Installer standards.

When testing an installation package against a best practice ACE, there is no target package involved. The source installation package is tested to see if it meets the standards defined in the ACEs.

The ConflictSolver best practice ACEs are the following:

Table 2-4 • Best Practice ACEs

ACE Rule	Description	Result	CARD Resolution
ACE04	Checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the CreateFolder table.	If a component does not have a key file, and if the directory/ component pair is not listed in the CreateFolder table, ACE04 fails.	A CreateFolder entry is created for the component.
ACE05	Checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package.	If more than one executable exists in a component, ACE05 fails.	Modifies the component so that only one EXE or DLL exists, and it adds new components for remaining EXE, DLL, OCX, HLP, CHM, TLB, SYS, and DRV files.

Table 2-4 • Best Practice ACEs

ACE Rule	Description	Result	CARD Resolution
ACE06	Checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file.	If the executable module is not the key file, ACE06 fails.	The executable module is automatically made the key file.
ACE25	Checks the entries in the package's CustomAction table to identify any hard coded paths in the Target field.	If a package has a hard coded path in the Target field of the CustomAction table, a warning is generated.	Open the package file in InstallShield Editor and change the hard coded paths to relative paths in the CustomAction table.
ACE26	Checks the Merge Modules in a package to confirm that they have been imported into the Application Catalog.	If a package refers to a Merge Module that does not exist in the Application Catalog, a warning is generated.	Import the missing Merge Module into the Application Catalog.
ACE27	Checks to see if data exists in the DuplicateFile MSI table that is not being executed with an associated DuplicateFiles action.	If a package contains data in the DuplicateFile table but not the necessary actions to use this data, a warning is generated.	Consider whether a DuplicateFiles or RemoveDuplicateFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the DuplicateFile table.
ACE28	Checks the Value field of the package's Environment table to identify any hard coded paths.	If any hard coded paths exist in the Value field of a package's Environment table, a warning is generated.	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Environment table to relative paths.
ACE29	Checks the Value field of a package's IniFile table to identify any hard coded paths.	If the Value field of a package's IniFile table has any hard coded paths, a warning is generated.	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the IniFile table to relative paths.

Table 2-4 • Best Practice ACEs

ACE Rule	Description	Result	CARD Resolution
ACE30	Checks to identify components with different ComponentIDs installing the same key file to the same directory. (A check for KeyPath conflicts across components.)	If components with different ComponentIDs installing the same key file to the same directory is identified, a warning is generated.	If a product component is installing a KeyPath file which is also being installed by a product with a different Component Id, the KeyPath ACE fails.
ACE31	Checks to see if data exists in the MoveFile MSI table that is not being executed with an associated MoveFiles action.	If a package contains data in the MoveFile table but not the necessary actions to use this data, a warning is generated.	Consider whether a MoveFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the MoveFile table.
ACE32	Checks entries in the Registry table to identify any hard coded paths in its Value field.	If there are any hard coded paths in the Value field of the package's Registry table, a warning is generated.	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Registry table to relative paths.
ACE33	Checks to see if data exists in the RemoveFile MSI table that is not being executed with an associated RemoveFiles action.	If a package contains data in the RemoveFile table but not the necessary actions to use this data, a warning is generated.	Consider whether a RemoveFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveFile table.

Table 2-4 • Best Practice ACEs

ACE Rule	Description	Result	CARD Resolution
ACE34	Checks to see if data exists in the RemoveIniFile MSI table that is not being executed with an associated RemoveIniFiles action.	If a package contains data in the RemoveIniFile table but not the necessary actions to use this data, a warning is generated.	Consider whether a RemoveIniFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveIniFile table.
ACE35	Checks to see if data exists in the RemoveRegistry MSI table that is not being executed with an associated RemoveRegistryValues action.	If a package contains data in the RemoveRegistry table but not the necessary actions to use this data, a warning is generated.	Consider whether a RemoveRegistryValues action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveRegistry table.
ACE36	Checks to see if a package's required Merge Modules have been imported into the Application Catalog.	If any of a package's required Merge Modules do not exist in the Application Catalog, a warning is generated.	Import the package's missing Merge Modules into the Application Catalog.

Resolving Best Practice ACE Errors

On the [Best Practices Page](#) of the Process Assistant, after you click **Detect**, any conflicts that are found are listed in the Best Practice ACE list. To resolve these conflicts, select the conflicts that you would like to resolve and click **Resolve**.

You can also resolve Best Practice ACEs conflicts along with standard ACE conflicts on the ConflictSolver [Conflicts View](#).






Task: *To resolve conflicts on the Conflicts View:*

1. From the **Product View**, select the product for which you want to view identified conflicts.
2. Select the **Conflicts View** for the product.
3. Select the category containing the automatically resolvable conflict from the **Conflict Information** list.
4. Expand the ACE containing conflicts that can be resolved.

The icon displayed next to the conflict indicates its status:

Table 2-5 •

Icon	Status
	Resolvable
	Resolved
	Unresolvable

5. Select the conflict you want to resolve and click **Resolve**. Alternatively, click **Resolve All** to resolve all resolvable conflicts for the selected conflict category.
6. For CARDS that can be resolved in multiple ways, and if the **Maximize user flexibility in selecting resolutions** option is selected on the **Resolution** tab of the **Options** dialog box, the **CARD Resolution Options** dialog box appears, prompting you to make a selection to determine how conflict resolution will proceed. Make a selection and click **OK** to continue.
7. If the **Preview and approve all resolutions** option on the **Resolution** tab of the **Options** dialog box is selected, the **Resolution Details** dialog box will appear, listing the proposed CARD resolutions that will be made to the package. Select the CARD resolutions that you want to resolve, and click **Approve** to execute them.

If the **Preview and approve all resolutions** option is not selected, the **Resolution Details** dialog box will not appear, and all selected conflict resolutions will be executed.

Configuring Conflict Detection

You can specify which ACEs you want to be selected by default on the **Conflict Options** panel of the **Conflict Wizard**.



Task: *To select the default ACEs:*

1. Select **Options** from the ConflictSolver **Tools** menu. The **Options** dialog box opens.
2. Click the **Conflicts** tab.
3. Select the default conflicts you want to check for from the **Predefined and Custom ACE Rules** area. Conflict rules (ACEs) associated with unselected boxes will not be performed by default during conflict identification. However, you can override these in the **Conflict Options** panel of the **Conflict Wizard**.
4. Click **OK**.

Configuring Resolution Options

On the **Resolution Tab** of the ConflictSolver **Options** dialog box, you can specify options to determine how conflict resolutions are performed.

Table 2-6 • Conflict Resolution Options

Option	Description
Perform resolutions against a transform file	<p>If this option is selected, ConflictSolver will, by default, save conflict detection resolutions in a transform file (MST). The original Windows Installer package (MSI) remains untouched.</p> <p>You can override this option for an individual package by going to that package's Conflict View and clicking Edit and change this option on the Resolution Options dialog box.</p>
Preview and approve all resolutions	<p>Select this option if you want to review and approve the suggested resolution for each conflict before ConflictSolver performs any resolutions. When the conflict detection process is completed, the Resolutions Details dialog box is displayed, listing the conflicts that were found and the CARDS that could be used to resolve these conflicts. Before ConflictSolver will continue resolving these conflicts, each conflict resolution must be approved. See Conflict Resolution Process for more information.</p>
Maximize user flexibility in selecting resolutions	<p>Some CARDS can resolve a conflict in more than one way. For example, ACE06 checks whether the executable module within the component is the key file, and accumulates a list of potential files to be used for the key file. If this option is selected, this list of files is displayed on the CARD Resolution Options dialog box, and you are prompted to select which file to use. If this option is not selected, the CARD will perform its default resolution.</p>

Detect Conflicts Page

On the Detect Conflicts Page, you will check the selected package against target packages to determine if conflicts would occur if all of these packages were installed on the same system.



Task: *To perform conflict detection:*

1. In the ConflictSolver Process Assistant, click the **Detect Conflicts** icon in the navigation bar. The **Detect Conflicts Page** opens.
2. In the Group tree, select the target packages that you would like to include in the conflict detection process.
3. Click **Detect**. Results appear in the **Conflict** tab of the **Output** window.

Target Packages

In conflict detection, target packages are the individual packages or groups of packages that you want to compare the source package(s) against. The source package is the package you are checking for conflicts.

Typically target packages are a group of applications and operating systems that are installed on workstations at your organization. You would perform conflict detection against these target packages to determine if any conflicts would occur if you installed the source package(s) on the same machine.

You select the target packages that you want to use in conflict detection on the **Target Information Panel** of the Conflict Wizard.

Configuring Conflict Detection

You can specify which ACEs you want to be selected by default on the **Conflict Options Panel** of the **Conflict Wizard**.



Task: *To select the default ACEs:*

1. Select **Options** from the ConflictSolver **Tools** menu. The **Options** dialog box opens.
2. Click the **Conflicts** tab.
3. Select the default conflicts you want to check for from the **Predefined and Custom ACE Rules** area. Conflict rules (ACEs) associated with unselected boxes will not be performed by default during conflict identification. However, you can override these in the **Conflict Options Panel** of the **Conflict Wizard**.
4. Click **OK**.

Resolve Conflicts Page

On the Resolve Conflicts Page, you will view all detected conflicts and choose to resolve any conflicts that have an automatic resolution.



Task:

To resolve conflicts:

1. In the ConflictSolver Process Assistant, click the **Resolve Conflicts** icon in the navigation bar. The **Resolve Conflicts Page** opens.
2. For the **Select the type of conflicts to display** option, select **Automatically resolvable conflicts**.
3. Select the ACE errors that you would like to automatically resolve.
4. Click **Resolve**. The **Resolution Details** dialog box opens prompting you to approve the changes.
5. Click **Approve**. The resolved conflicts now are listed with a check mark icon.
6. You can also view the manually resolvable conflicts. See [Application Conflict Evaluators \(ACEs\)](#) for resolution instructions.

Automatic and Manual Conflict Resolution

Automatic Conflict Resolution

To identify conflicts between multiple installation packages, ConflictSolver uses Application Conflict Evaluators (ACEs). Some ACE failures can be resolved automatically by ConflictSolver. These automatic resolution techniques are defined in Conflict Application Resolution Definition (CARD) rules. ACEs that can be automatically resolved by CARDs are referred to as Best Practice ACEs: ACE02, ACE04, ACE05, ACE06, ACE07, ACE15, ACE18, ACE19, ACE20. CARDs resolve errors found by the corresponding ACE of the same number.

You can choose to automatically resolve ACEs on the **Conflicts View** for the product. Select the conflict you want to resolve and click **Resolve**. Alternatively, click **Resolve All** to resolve all resolvable conflicts for the selected conflict category. See [Resolving Best Practice ACE Errors](#)

Manual Conflict Resolution

Due to their complexity, some conflicts require manual resolution using Tuner or InstallShield Editor.

For information on manual solutions, consult the [Application Conflict Evaluators \(ACEs\)](#) topic or individual ACE topics. These topics can provide valuable information regarding what is causing the conflict, so you can use Tuner or InstallShield Editor to solve it.

Once you have resolved detected conflicts manually, you can re-import the package into ConflictSolver.

Configuring Resolution Options

On the Resolution Tab of the ConflictSolver **Options** dialog box, you can specify options to determine how conflict resolutions are performed.

Table 2-7 • Conflict Resolution Options

Option	Description
Perform resolutions against a transform file	<p>If this option is selected, ConflictSolver will, by default, save conflict detection resolutions in a transform file (MST). The original Windows Installer package (MSI) remains untouched.</p> <p>You can override this option for an individual package by going to that package's Conflict View and clicking Edit and change this option on the Resolution Options dialog box.</p>
Preview and approve all resolutions	<p>Select this option if you want to review and approve the suggested resolution for each conflict before ConflictSolver performs any resolutions. When the conflict detection process is completed, the Resolutions Details dialog box is displayed, listing the conflicts that were found and the CARDS that could be used to resolve these conflicts. Before ConflictSolver will continue resolving these conflicts, each conflict resolution must be approved. See Conflict Resolution Process for more information.</p>
Maximize user flexibility in selecting resolutions	<p>Some CARDS can resolve a conflict in more than one way. For example, ACE06 checks whether the executable module within the component is the key file, and accumulates a list of potential files to be used for the key file. If this option is selected, this list of files is displayed on the CARD Resolution Options dialog box, and you are prompted to select which file to use. If this option is not selected, the CARD will perform its default resolution.</p>

Reporting Page

On the Reporting Page, you can generate package reports listing data about conflicts, extended attributes, files, and registry entries.

Select the report that you would like to generate, and click **Generate**. The report opens in a new browser window. The following reports are available:

Table 2-8 • Available Reports



Report	Description
Package	<p>Provides a report including the following information:</p> <ul style="list-style-type: none">● Product Information—Lists all information that appears on the Product View for a package.● Conflict Information—Lists all conflicts found during the most recent Conflict Wizard execution, or the most recent Import (if conflict identification was performed).● Validation Information—Lists all validation errors and warnings found during the most recent validation execution for that package.  <p>Note • Validation information is persisted only for validations performed after the package was imported into the Application Catalog; validation results for validation performed during import are not persisted.</p> <ul style="list-style-type: none">● Extended Attributes—Lists user-defined information that appears on the Extended Attributes View for a package.
File	<p>The following information is displayed:</p> <ul style="list-style-type: none">● Product Information—Lists all information that appears on the Product View for a package.● Files—Lists all of the files that are included with the package. These files are also listed on the Files/Components View for that package.
Registry	<p>The following information is displayed:</p> <ul style="list-style-type: none">● Product Information—Lists all information that appears on the Product View for a package.● Registry—Lists any registry entries created or changed by the package. These registry entries are also listed on the Registry View for that package.

Table 2-8 • Available Reports (cont.)

Report	Description
Crystal Reports	<p>The following reports in Crystal Report format are available:</p> <ul style="list-style-type: none"> • Crystal Reports Conflicts—Provides a report of all conflicts found during the most recent Conflict Wizard execution, or the most recent Import (if conflict identification was performed). This report is also available on a per-product basis by right-clicking on a product in the Product View. • Crystal Reports Files—Provides a comprehensive listing of all files in ConflictSolver, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View. • Crystal Reports Registry—Provides a comprehensive listing of all registry entries in ConflictSolver, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View.  <p>Note • The three Crystal Reports selections will be listed on the Report menu if the Display RPT Reports option on the General tab of the ConflictSolver Options dialog box is selected. To suppress the display of the Crystal Reports selections, unselect this option.</p>



Note • You can also generate reports by selecting the report name from the ConflictSolver Reports menu.

Organize Package Page

On the Organize Package Page, you select the Group(s) where you want the imported package to belong to, and you can also choose to add the file to the Software Repository.

About the “Draft” Group

When you import a package into the Application Catalog using the ConflictSolver Process Assistant, that package is automatically imported into a special group called **Draft**, a temporary working group that isolates an imported package that has not yet been checked for conflicts from the rest of the packages in the Application Catalog. The **Draft** group is only visible when you select **Current** on the **Select a Package Page** of the Process Assistant.

By placing these newly imported packages into the **Draft** group, you are indicating that these packages are in the process of being prepared for distribution, and should not be used as target packages in conflict detection. After you have performed validation, conflict detection, and conflict resolution on a Draft package, and it is ready for distribution at your organization, you should then move it out of the **Draft** group using the **Organize Package Page**. The package will then be visible on the ConflictSolver and Application Manager **Product View**.

About the Software Repository

A Windows Installer package is made up of many files that are executed when the setup is run. However, only the .msi file is imported into the Application Catalog database. To safeguard these additional files against alteration or being misplaced, you can choose to manage these files using the **Software Repository**. When you import a package into the Software Repository, all of its associated installation files are stored in the Software Repository location defined for the open Application Catalog.

If you want to add a file you are importing to the Software Repository, select the **Add package to the Software Repository** option on the **Organize Package Page**. In order to select this option, you must be connected to an Application Catalog that has the Software Repository enabled. See [Managing Application Catalog Databases](#) and [Enabling the Software Repository in an Existing Application Catalog](#).



Task: **To relocate a package within your Application Catalog:**

1. In the ConflictSolver Process Assistant, click the **Organize** icon in the navigation bar. The **Organize Package Page** opens.

In the **Destination Groups** list, the check boxes next to the Groups that your package is in (if any) are selected. If you have just imported a package, it is still in the **Draft** group, which is not displayed on this list.

2. Select the check boxes next to the **Group(s)** where you want this package to be located.
3. If you want to add this package to the Software Repository, select the **Add package to the Software Repository** option.



Note • The **Add package to the Software Repository** option is only displayed if you are connected to an Application Catalog that has the Software Repository enabled. See [Enabling the Software Repository in an Existing Application Catalog](#).

4. Click **Organize**. The package is moved into the selected group(s) and, optionally, copied into the Software Repository.

Current and New Packages

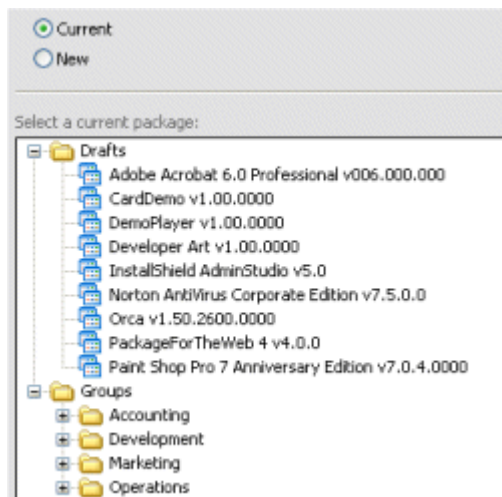
On the **Select a Package Page**, you choose the package that you are going to process using the ConflictSolver Assistant. You can choose to use either a **Current** or **New** package:

- **Current**—Package that has already been imported into the Application Catalog.
- **New**—Package that will be imported into the Draft Group of the Application Catalog, a working Group used for packages in the process of being prepared for distribution.

After you have selected a package, you can use it in the **Validation**, **Best Practices**, **Detect Conflicts**, **Resolve Conflicts**, **Reporting**, and **Organize Package** tasks of the ConflictSolver Assistant.

Draft Packages

When you import a package into the Application Catalog using the ConflictSolver Process Assistant, that package is automatically imported into a special group called Draft, a temporary working group that isolates an imported package that has not yet been checked for conflicts from the rest of the packages in the Application Catalog. The **Draft** group is only visible when you select Current on the [Select a Package Page](#) of the Process Assistant.



By placing these newly imported packages into the **Draft** group, you are indicating that these packages are in the process of being prepared for distribution, and should not be used as target packages in conflict detection.

After you have performed validation, conflict detection, and conflict resolution on a Draft package, and it is ready for distribution at your organization, you should then move it out of the Draft group using the [Organize Package Page](#) of the Process Assistant. The package will then be visible on the ConflictSolver and Application Manager **Product View**.



Note • While it is possible to move a package out of the Draft group before conflict detection and resolution have been performed, it is not recommended.

Creating a New Group

You can create a new group using the New button on the [Organize Package Page](#) of the Process Assistant.



Task: *To create a new group:*

1. If a package is not currently selected, go to the [Select a Package Page](#) and select a package.
2. Click on the **Organize Package** button in the navigation bar. The [Organize Package Page](#) opens.
3. Click **New**. The **New Group Properties** dialog box opens.
4. Enter a **Name**, **Description** and any desired **Comments** for the new group and click **OK**. The new Group is listed in the group tree and is automatically selected.

Test Page



Edition • *QualityMonitor is included with AdminStudio Professional and Enterprise Editions.*

On the Test Page, you test the package installation on the targeted deployment environment to ensure that it works as expected.



Task: *To test a package installation:*

1. Click on the **Test** icon in the navigation bar. The **Test Page** opens.
2. In the **Package File** field, specify the name of the package that you want to test. The fields on the Test page change depending upon whether the selected package is installed on your machine.
 - **If the selected application is already installed on your machine**, skip to step 6.
 - **If the selected application is not installed on your machine**, you are prompted to install the application.
3. Include any additional transform files in the installation by adding them to the **Transform List**.
4. Click **Install** to install the package. The package's installation program runs.
5. Complete the steps required to install the selected package. When the installation is complete, the Test selection list appears.
6. From the Test selection list, select the tests that you want to run:
 - **Execute all deployment**—Deployment tests are performed against the installed product, ensuring that the product has been installed correctly and all key functionality works in the installed environment. See [Deployment Testing](#)





- **Verify the package in a lockdown environment**—You are provided with a list of available shortcuts in the package and all of the executables in the package. You can then launch via the shortcut or executable, and exercise functionality in the application. When you close the application, information about the executable is listed, allowing you to see failures in the application execution. See [Lockdown and Runtime Testing](#)
- **Verify the package installation status using MSI Doctor**—Allows you to verify if an MSI package is installed properly. This helps prevent users from seeing an auto-repair dialog box when they run the application. See [Using MSI Doctor to Verify Package Deployment Status](#).
- Click **Test** to run the selected test against the package.

Distribute Page

On the **Distribute Page**, you prepare your package for distribution.



Task: *To prepare your package for distribution:*


1. Click on the **Distribute** icon in the navigation bar. The **Distribute Page** opens.
2. Click **Browse** to locate and select the package.
3. Should that package require transforms, click the **New** button () to launch the **Select Transform Files** dialog box and select a transform file.
 - If the package requires multiple transforms, you can repeat the procedure as necessary.
 - The order in which transforms are applied can be changed by selecting a transform and clicking the **Move Up** () and **Move Down** () buttons.
 - Delete a transform file from the Transforms list by selecting the transform and clicking the **Delete** button ().
4. From the **Distribution Type** list, select the type of distribution you want to perform. A description of each type appears when it is selected.
5. Click **Distribute** to complete the distribution process.




See Also

[Preparing Packages for Distribution Using Distribution Wizard](#)
[Distributing Transforms With an MSI Package](#)

Distributing Transforms With an MSI Package

To distribute a package that requires transforms, you would specify the files that you want to include in the **Transform Files** list on the **Distribute Page**.

After you have selected the package to distribute, click the **New** button () to launch the **Select Transform Files** dialog box, and select a transform file (.mst).

- If the package requires multiple transforms, you can repeat the procedure as necessary.
- The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up () and Move Down () buttons.
- Delete a transform file from the Transforms list by selecting the transform and clicking the Delete button ()

Using the AdminStudio Interface

The AdminStudio interface is the central application for AdminStudio. From it, you can launch AdminStudio tools, launch the Process Assistant, create workflow templates and projects, and connect to and create Application Catalogs. You can launch the AdminStudio interface from the Windows Start menu.



Note • The Process Assistant is a procedure-based alternative interface to the standard AdminStudio interface. For more information, see [Using the Process Assistant](#).

AdminStudio interface documentation is presented in the following sections:

Table 3-1 • AdminStudio interface Documentation

Section	Description
Configuring the AdminStudio Interface	Includes information on customizing the AdminStudio Interface, including setting shared directories.
Working with Tools	Explains how to configure AdminStudio tools and also external tools necessary to perform workflow tasks.
Workflows and Projects	Explains how to use workflow functionality to define repeatable processes to accomplish your goals.
Frequently Asked Questions	A list of questions frequently asked by AdminStudio users, with links to the appropriate help topics.
AdminStudio Interface Reference	This section contains an exhaustive description of each dialog box, Wizard, and UI element in AdminStudio.

Configuring the AdminStudio Interface

This section explains how to configure and customize the AdminStudio interface. The following topics are discussed:

Table 3-2 • Topics on AdminStudio Interface Configuration

Category	Topics
Launching Applications	<ul style="list-style-type: none">• Launching AdminStudio Applications
Configuring Application Catalog Settings	<ul style="list-style-type: none">• Specifying the Shared AdminStudio Application Catalog• Specifying the AdminStudio Shared Location
Setting Interface Preferences	<ul style="list-style-type: none">• Setting the Workflow Task Help Page Location• Configuring How Often AdminStudio Checks for Updates• Configuring AdminStudio to Stay on Top• Generating a Debug Log for AdminStudio
Configuring Interactivity Settings	<ul style="list-style-type: none">• Specifying the AdminStudio Enterprise Server URL• Specifying the Configuration Manager Web Console URL

Launching AdminStudio Applications

Individual AdminStudio applications, such as ConflictSolver, InstallShield Editor, or Repackager, can be launched by clicking on the tool in the **Tools Gallery** on the [AdminStudio Start Page](#).

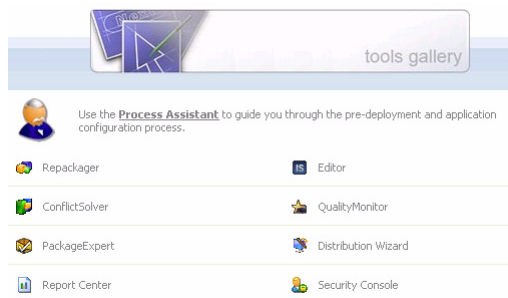


Figure 3-1: Tools Gallery on Start Page

You can also launch AdminStudio applications by double-clicking on the tool icon in the Tools Gallery on the [Tools Tab](#).

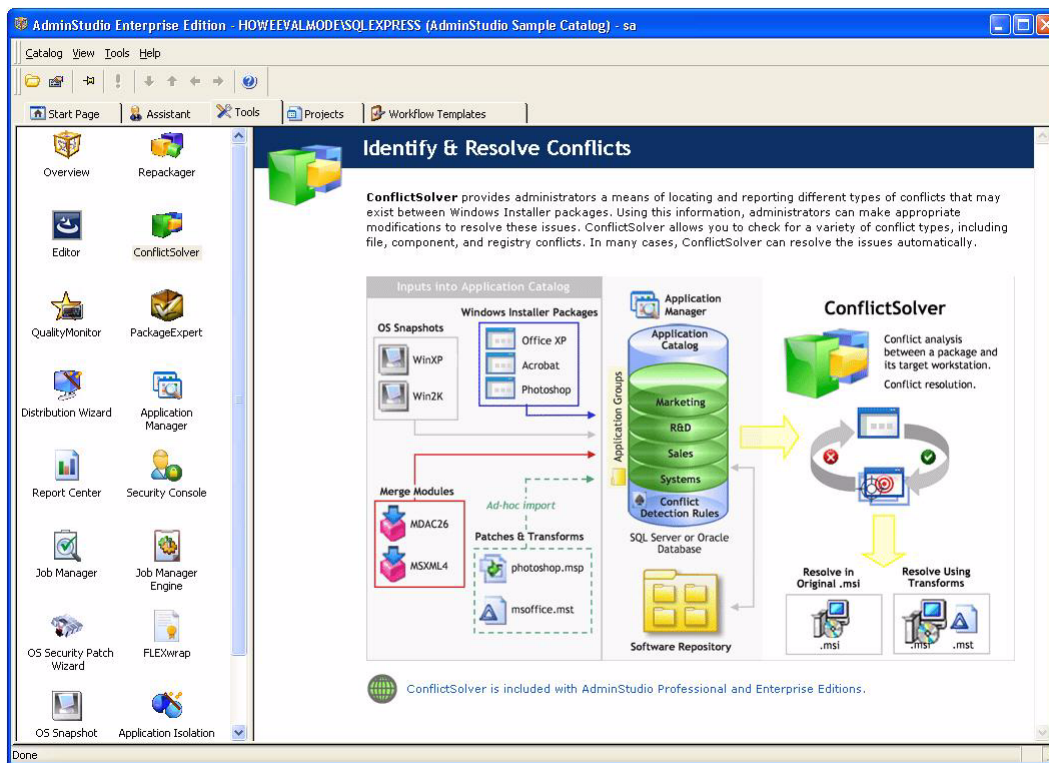


Figure 3-2: Tools Gallery on the Tools Tab



Note • If a core AdminStudio tool is not in the Tools Gallery or available from the menu, it may mean that either you are not assigned to a Role that has permission to use that Tool, or that the Tool is not available in your Edition of AdminStudio.



Note • You can also launch AdminStudio tools from the Process Assistant. See [Using the Process Assistant](#) for more information.

Specifying the Shared AdminStudio Application Catalog

This field displays the name and location of the shared AdminStudio Application Catalog. Click the Browse button to connect to another Application Catalog using the Connect to Application Catalog dialog box.



Task: *To specify the location of the AdminStudio Shared Location:*

1. Launch the AdminStudio interface.
2. From the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. Select the **Application Catalog** tab.
4. Select the **Shared AdminStudio Application Catalog**.
5. Click **OK** to close the **Options** dialog box.

Specifying the AdminStudio Shared Location

The AdminStudio Shared Directory (also referred to as the AdminStudio Shared Location) contains shared information for repackaging and conflict identification, and other AdminStudio functions. The AdminStudio Shared Directory contains the following:

- The Shared AdminStudio.ini file, which specifies default Application Catalog database settings
- Application Manager duplicate package identifier options
- Repackager isrepackager.ini exclusion list
- OS Snapshot issnapshot.ini file
- ConflictSolver user-defined ACEs
- Distribution Wizard Distribution Type templates and .ini files

If you are working in a team environment, the AdminStudio Shared Directory should be set to a centralized network location, accessible by all AdminStudio users at your organization, rather than on your local machine. Follow the steps below to specify the location of the AdminStudio Shared Directory.



Note • To maintain consistency when creating workflows, InstallShield recommends that you set the AdminStudio Shared Directory the same for each AdminStudio user.



Task: *To specify the location of the AdminStudio Shared Directory:*

1. Launch the AdminStudio Interface.
2. From the **Tools** menu, select **Options**. The **Options** dialog box opens.

3. In the **Options** dialog box, select the **Locations** tab.
4. Enter or browse to the directory for the **AdminStudio Shared Location**.
5. Click **OK** to close the **Options** dialog box.

Setting the Workflow Task Help Page Location

The **Task Help Page Location** is the directory where you want to store all HTML pages that serve as workflow task instructions.



Task: *To specify the location of task help pages:*

1. Launch the AdminStudio interface.
2. From the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. Select the **Locations** tab.
4. Enter or browse to the **Task Help Page Location**, the location where workflow task help pages are stored.
5. Click **OK** to close the **Options** dialog box.

Configuring How Often AdminStudio Checks for Updates

You specify how often you want AdminStudio to check for updates on the **Updates** tab of the **Options** dialog box.



Task: *To configure how often AdminStudio checks for updates:*

1. Open AdminStudio.
2. On the **Tools** menu, click **Options**. The **Options** dialog box opens.
3. On the **Updates** tab, select the frequency that AdminStudio will check for software updates:
 - Once every 15 days
 - Once every 30 days
 - Once every 60 days
 - Never
4. Click **OK** to close the **Options** dialog box.

Configuring AdminStudio to Stay on Top

When you launch AdminStudio tools, you can specify whether they open in front of or behind the AdminStudio interface. If you select **Always on Top** from the **View** menu, application will always open behind the AdminStudio interface.



Task: *To configure AdminStudio to stay on top of other applications:*

1. Launch the AdminStudio interface.
2. From the **View** menu, select **Always on Top**.

Generating a Debug Log for AdminStudio

To create a debug log for AdminStudio, perform the following steps.



Task: *To generate a debug log for AdminStudio:*

Use the following registry value to turn debugging on. Once this debugging is turned on, a log file will be created in the same location as the .exe file.

`[HKLM\Software\InstallShield\AdminStudio] DebugLogLevel="3"`

Levels 0, 1, 2, 3 are supported with 3 being the highest. Default is level 0.

Specifying the AdminStudio Enterprise Server URL

You specify the Web site URL of AdminStudio Enterprise Server on the **Application Catalog** tab of the **Options** dialog box.



Task: *To specify the AdminStudio Enterprise Server URL:*

1. Launch the AdminStudio interface.
2. On the **Tools** menu, click **Options**. The **Options** dialog box opens.
3. Select the **Application Catalog** tab.
4. Enter the **AdminStudio Enterprise Server URL**.
5. If you have more than one AdminStudio Enterprise Servers, select the **Make this the default Shared AES URL** to make the URL in the **AdminStudio Enterprise Server URL** field the default.
6. If you want to **Allow automatic authentication** by AdminStudio Enterprise Server, select this option.
7. Click **OK** to close the **Options** dialog box.

Specifying the Configuration Manager Web Console URL

AdminStudio 9.5

Enter the URL for the Configuration Manager Web Console Web site. You specify the Web site URL of the Configuration Manager Web Console on the **Locations** tab of the **Options** dialog box.



Task: *To specify the Configuration Manager Web Console URL:*

1. Launch the AdminStudio interface.
2. On the **Tools** menu, click **Options**. The [Options Dialog Box](#) opens.
3. Select the **Locations** tab.
4. Enter the **Configuration Manager Web Console URL**.
5. Click **OK** to close the **Options** dialog box.

Working with Tools

Topics in this section involve adding, configuring, and associating tools in the **Tools Gallery** on the [Tools Tab](#). This section contains the following topics:

- [Adding New Tools to the Tools Gallery](#)
- [Editing Properties for an Existing Tool](#)
- [Adding Command-Line Configurations for an Existing Tool](#)
- [Modifying Command-Line Configurations for an Existing Tool](#)
- [Deleting Command-Line Configurations from an Existing Tool](#)
- [Associating Tools with Tasks](#)
- [Running Associated Tools in Projects](#)
- [Deleting Existing Tools](#)
- [Limiting Tool Accessibility](#)

Adding New Tools to the Tools Gallery



Task: *To add a new tool to the Tools Gallery:*

1. From the **Tools Tab**, right-click in the Tools Gallery and select **Add Tool**. The **Add Tool Wizard** opens.
2. On the **Welcome Panel**, click **Next**. The **Tool Properties Panel** opens.
3. On the **Tool Properties Panel**, provide the necessary details about the tool. Click **Next**. The **Command-Line Configuration Panel** opens.
4. If there are command-line options you want associated with the tool, do so from the **Command-Line Configurations Panel**. Each tool can have multiple command-line options associated with it for different purposes.
5. Click **Finish**.

After using the **Add Tool Wizard**, the new tool appears in the Tools Gallery and is available for use in workflows and projects.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Editing Properties for an Existing Tool

To edit the properties of a Tool in the Tools Gallery, perform the following steps.



Task: *To edit an existing tool's properties:*

1. From the **Tools Tab**, right-click the tool in the Tools Gallery and select **Properties**. The **Tool Properties** dialog box is displayed.
2. Click the **Properties** tab.
3. Modify tool properties as necessary.
4. Click **OK** to apply the changes.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Adding Command-Line Configurations for an Existing Tool

To add command-line configurations for an existing Tool, perform the following steps:



Task: *To add a command-line configuration to an existing tool:*

1. From the **Tools Tab**, right-click the tool to which you want to add a configuration from the Tools Gallery and select **Properties**. The **Tool Properties** dialog box is displayed.
2. Click the **Configuration** tab.
3. Click **Add**. The **Command-Line Properties** dialog box is displayed.
4. Enter a description and the command-line configuration.
5. Click **OK** to close the **Command-Line Properties** dialog box.
6. Click **OK** to close the **Tool Properties** dialog box.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Modifying Command-Line Configurations for an Existing Tool

To modify command-line configurations for an existing Tool, perform the following steps:



Task: *To edit and existing tool's command-line configurations:*

1. From the **Tools Tab**, right-click the tool in the Tools Gallery and select **Properties**. The **Tool Properties** dialog box is displayed.
2. Click the **Configuration** tab.
3. Select the command-line configuration you want to edit and click **Modify**. The **Command-Line Properties** dialog box appears.
4. Modify the description and/or command line.
5. Click **OK** to dismiss the **Command-Line Properties** dialog box.
6. Click **OK** in the **Tool Properties** dialog box to apply the changes.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Deleting Command-Line Configurations from an Existing Tool

To delete command-line configurations from an existing Tool, perform the following steps.



Task: *To delete a command-line configuration from an existing tool:*

1. From the **Tools Tab**, right-click the tool from which you want to remove the configuration in the Tools Gallery and select **Properties**. The **Tool Properties** dialog box is displayed.
2. Click the **Configuration** tab.
3. Select the configuration you want to remove and click **Delete**.
4. Click OK to close the **Tool Properties** dialog box.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Associating Tools with Tasks

To associate a Tool with a Workflow task, perform the following steps.



Task: *To associate a tool with a task:*

1. Click the **Workflow Templates** tab in the Interface.
2. In the **Workflows** tree, expand a Workflow to display all of its tasks.
3. From the **Workflows** tree, select the task with which you want to associate the tool. The **Task Properties** are displayed.
4. From the **Tool** list, select the tool you want to associate with the task. If the necessary tool is not listed, select **<New Tool...>** from the list to add the tool.
5. In the **Tool Configuration** list, select the configuration you want to use with the tool. If the configuration is not listed, click **Configure** to create the new configuration.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Running Associated Tools in Projects

To run associated Tools in Projects, perform the following steps.



Task: *To run a tool associated with a task:*

1. Select the **Projects** tab from the Interface.
2. Expand a Project in the **Projects** tree to display all of its tasks.
3. Right-click the task with which the tool is associated and select **Run Task** from the context menu.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Deleting Existing Tools

To delete a Tool from the Tools Gallery, perform the following steps.



Task: *To delete an existing tool from the Tools Gallery:*

1. From the **Tools Tab**, right-click the tool you want to delete from the Tools Gallery and select **Delete**.
2. Confirm the deletion.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Limiting Tool Accessibility



Edition • AdminStudio Enterprise Server is included in AdminStudio Enterprise Edition.

Accessibility of tools is determined by a user's assigned Roles. If you want to modify an existing Role so that users assigned to that Role can no longer access a specific tool, see [Editing an Existing User Role](#) in the [Managing Roles and Permissions](#) section.

Workflows and Projects

Workflows, which can be created and modified from the **Workflow Templates** tab, are the basis for all projects in AdminStudio. These workflows consist of defined tasks, with which instructions (in the form of HTML files) and tools can be associated. Users can then create projects based on these workflows, and execute them—following the specific steps defined in the workflow. This allows you to create specific, repeatable procedures to accomplish your application migration goals.



Tip • If you update a workflow, all projects based on that workflow will reflect the changes made to the workflow.

The following topics relate to workflows and projects:

- [Creating and Editing Workflows](#)
- [Creating and Using Projects](#)
- [Saving Workflow and Project Changes](#)
- [Workflow Project Example: Using the New Workflow Project Wizard](#)
- [Workflows, Projects, and Permissions](#)

Creating and Editing Workflows

Workflows serve as templates upon which projects are based. Typically, only a few individuals create workflows, while others create projects and execute the projects to accomplish the workflow goal.

The following topics relate to creating and executing workflows:

- [Creating New Workflows](#)
- [Renaming Workflows](#)
- [Filtering Workflows](#)
- [Deleting Workflows](#)
- [Creating New Tasks](#)
- [Modifying Task Properties](#)

- [Creating Notes for a Task](#)
- [Renaming Tasks](#)
- [Reordering Tasks](#)
- [Associating Help Files with Tasks](#)
- [Deleting Tasks](#)
- [Adding New Tools from the Workflow Templates Tab](#)

Creating New Workflows



Task: *To create a new workflow:*

1. Click the **Workflow Templates** tab in the Interface. The Workflow Templates tab appears.
2. Right-click in the Workflows tree pane and select New Workflow. A new Workflow is listed.
3. Provide a name for the workflow.

Renaming Workflows



Task: *To rename an existing workflow:*

1. Select the **Workflow Templates** tab in the Interface.
2. Right-click the workflow you want to rename and select Rename from the context menu.
3. Provide a new name for the workflow.

Filtering Workflows



Task: *To display a specific workflow:*

1. From the Interface, click the **Workflow Templates** tab.
2. From the drop-down menu above the Workflows tree, select the workflow you want to display.

Deleting Workflows



Task: *To delete an existing workflow:*

1. Click the **Workflow Templates** tab in the interface.
2. From the Workflows tree, right-click the workflow you want to delete and select Delete from the context menu.
3. Confirm the deletion by clicking Yes in the resulting dialog box.

Creating New Tasks



Task: *To create a new task:*

1. Select the **Workflow Templates** tab in the interface.
2. Right-click the workflow to which you want to add the task and select New Task. Alternatively, right-click on a task and select New Task to create a subtask.

A new task appears named NewTask nn , and the Task Properties view for that task is displayed.

3. Enter a name for the new task.
4. Modify properties for the task.

Modifying Task Properties



Task: *To modify properties for an existing task:*

1. Click on the **Workflow Templates** tab.
2. From the Workflows tree, select the task you want to modify. The Task Properties view for the selected task is displayed.
3. Change Task Properties as necessary for the task.

Creating Notes for a Task



Task: *To create notes for a task:*

1. Select the **Projects** tab in the Interface.
2. From the Projects tree, select the task to which you want to add notes. The Project Task Properties view appears for the selected task.
3. Enter notes in the Notes field.



Tip • You can also add notes to a task in the **Workflow Templates** tab. If you do, all projects based on that workflow will use the notes you enter as the default notes for the specific task.



Note • There is a 255 character limit on notes.

Renaming Tasks



Task: *To rename an existing task:*

1. Select the **Workflow Templates** tab in the Interface.
2. Right-click the task you want to rename and select Rename from the context menu.
3. Provide a new name for the task.

Reordering Tasks



Task: *To change the task order:*

1. Select the **Workflow Templates** tab in the Interface.
2. In the Workflows tree, select the task you want to move.
3. From the toolbar, click Move Up or Move Down to change the order in which tasks are performed. Click Move Right to make a task a subtask of another task; click Move Left to promote a task.
4. Repeat the previous steps as necessary.

Associating Help Files with Tasks



Task: *To associate a help file with a task:*

1. Click the **Workflow Templates** tab in the Interface.
2. From the Workflows tree, select the task with which you want to associate the help file. The Task Properties view appears for the selected task.
3. In the Help File field, enter the name and location of the help file, or click Browse and navigate to it.
 - The help file can either be local, or you can use a URL (for example, <http://www.mycompany.com/myURL.htm>).
 - You can also click the Edit HTML button to the right of the Browse button to open a default HTML page in an HTML editor as a starting point.



Note • Help files must be in HTML format.

Deleting Tasks



Task: *To delete an existing task:*

1. Select the **Workflow Templates** tab in the Interface.
2. In the Workflows tree, right-click the task you want to remove and select Delete.
3. From the resulting dialog box, click Yes to confirm the deletion.

Adding New Tools from the Workflow Templates Tab



Task: *To add a new tool to the Tools list from the Workflow Templates tab:*

1. Click the **Workflow Templates** tab in the Interface.
2. From the Workflows tree, select the task with which you want to associate the new tool. The Task Properties view appears for the selected task.
3. From the Tool list in the Task Properties view, select <Add Tool ...>. The Add New Tool dialog box opens.
4. In the **Add New Tool** dialog box, enter properties about the tool.
5. Click OK.

Creating and Using Projects

Projects, which are based on existing workflows, are the procedures followed to accomplish a set goal. Projects may include instructions describing what to do, and perhaps links to tools necessary to perform tasks. They also allow you to provide notes to help document issues that may arise during a project.



Tip • If you update a workflow, all projects based on that workflow will reflect the changes made to the workflow.

The following topics relate to creating and using projects:

- [Creating Workflows with the New Workflow Project Wizard](#)
- [Filtering Projects](#)
- [Executing Projects](#)
- [Running Associated Tools in Projects](#)
- [Deleting Projects](#)

Creating Workflows with the New Workflow Project Wizard



Task: *To create a workflow using the New Workflow Project Wizard:*

1. Launch AdminStudio.
2. Click the **Projects** tab.
3. Right-click in the Projects tree and select New Project. The **New Workflow Project Wizard** launches.
4. From the **Welcome Panel**, click **Next**. The **Workflow Selection Panel** appears.
5. From the **Workflow Selection Panel**, select the workflow on which you want to base the new project.
6. Provide a name for the new project and click **Next**. The Source Package panel appears.
7. From the **Source Package Panel**, specify the name and location of the source package used in this project. Alternatively, click Browse to navigate to it.
8. Click **Next**. The Target Directory and Filename panel appears.
9. From the **Target Directory and File Name Panel**, specify the Target Directory in which you want to store all files associated with this project.
10. In the Target File Name field, provide a name for the output file. Depending on the task being executed, the appropriate extension will be added to the file name.
11. Click Finish. The new Workflow is now listed.

Filtering Projects



Task: *To display a specific project:*

1. From the Interface, click the **Projects** tab.
2. From the drop-down menu above the Projects tree, select the project you want to display.

Executing Projects



Task: *To execute a project:*

1. From the Interface, click the **Projects** tab.
2. Display the project you want to execute. If you want to only display that project, use the filter above the Projects tree.
3. Click the first task in the project.
4. Perform the task.
5. When finished with the task, click the box to the left of the task.
6. Repeat for subsequent tasks in the project.

Running Associated Tools in Projects



Task: *To run a tool associated with a task:*

1. Select the **Projects** tab from the Interface.
2. Expand a Project in the Projects tree to display all of its tasks.
3. Right-click the task with which the tool is associated and select Run Task from the context menu.



Note • In AdminStudio, tools are any external application or file that you can launch from a workflow or project. This typically is an application, but can be a simple document or batch file necessary to completing the project.

Deleting Projects



Task: *To delete an existing project:*

1. Click the **Projects** tab in the Interface.
2. From the Projects tree, right-click the project you want to delete and select Delete Project from the context menu.
3. Confirm the deletion by clicking Yes in the resulting dialog box.

Saving Workflow and Project Changes

Because AdminStudio uses a database (the Application Catalog) to store information involving Workflows and Projects, all changes are stored immediately. There is no need to “save” your modifications; AdminStudio performs this automatically.

Workflow Project Example: Using the New Workflow Project Wizard

The following basic example covers creating a workflow and project which takes advantage of command-line functionality available in AdminStudio.

Prior to creating projects, you must create a workflow on which to base the project. This workflow might involve few steps, or it might cover as broad of a task as repackaging a legacy installation, editing it in InstallShield Editor, customizing it in Tuner, performing application isolation, identifying and resolving conflicts, distributing it, and entering information about it into a third-party tracking system.

In this example, you are going to create a basic workflow involving two steps: repackaging a legacy installation and opening the resulting InstallShield Editor project.

Creating a New Workflow



Task: *To create a new workflow:*

1. Launch AdminStudio.
2. Click the **Workflow Templates** tab in the Interface. The Workflow Templates tab appears.
3. Right-click in the Workflows tree pane and select New Workflow. A new Workflow is listed.
4. Name the workflow My Workflow Example.
5. Right-click My Workflow Example and select New Task.
6. Name the task Repackage a Legacy Setup.

7. With the Repackage a Legacy Setup task selected, from the Task Properties pane, select Repackager from the Tool menu. This associates Repackager with this task.
8. Change the Tool Configuration to Repackage a legacy setup. This associates a predefined command line (-app "[SourcePackage]" -o "[TargetDir]" -pp "[TargetFileName]") with this task.
9. Right-click My Workflow Example and select New Task.
10. Name the task Open the Repackaged Setup.
11. With Open the Repackaged Setup selected, from the Task Properties pane, select Editor from the Tool menu. This associates InstallShield Editor with this task.
12. Change the Tool Configuration to Edit an InstallShield Editor project created with Repackager. This associates a predefined command line (-"[TargetDir]\[TargetFileName].ism") with this task.

Creating a Project Based on the Workflow



Task: *To create a project based on your new workflow:*

1. Click the **Projects** tab.
2. Right-click in the Projects tree and select New Project. The **New Workflow Project Wizard** launches.
3. From the **Welcome Panel**, click **Next**. The **Workflow Selection Panel** appears.
4. From the **Workflow Selection Panel**, select My Workflow Example as the workflow on which you want to base the new project.
5. Name the new project My Sample Project and click **Next**. The Source Package panel appears.
6. From the **Source Package Panel**, click Browse and navigate to the following directory:
AdminStudio Installation Directory\Tutorial\Repackager\PackageForTheWeb 3
7. Select PackageForTheWeb3.exe and click Open. This value (the directory and package name) are stored in the SourcePackage variable, which is used by the command line in Repackager set when you created the workflow.
8. Click **Next**. The **Target Directory and File Name Panel** opens.
9. From the Target Directory and File Name panel, specify the directory where you want to store files associated with your project. For this example, use C:\AdminStudio Shared\Test\WorkflowExample This value is written to the TargetDir variable used in the command line set for InstallShield Editor when creating the workflow.
10. Set the Target File Name to WorkflowProjectEx. This value is written to the TargetFileName variable used in the command line set for InstallShield Editor when creating the workflow.
11. Click Finish. The new Workflow Project is now listed.

Running the Workflow



Task:

To run the workflow:

1. Expand the workflow My Sample Project in the Projects tree.
2. Right-click the task Repackage a Legacy Setup and select Run Task. <

When Repackager launches, it reads the value SourcePackage to determine the file to repackage. It also reads TargetDir and TargetFileName to determine where to place the output and what to call the output file.

3. After completing the steps above, right-click on the second step in the project and run the associated tool. InstallShield Editor launches, and opens the project file (.ism) created by Repackager.

Summary

This is just a brief example of how AdminStudio tools can be made aware of each other during a project. When crafting workflows, create command lines to streamline your projects.

Workflows, Projects, and Permissions

AdminStudio interface functionality (including workflows, projects, and the Tools Gallery) is directly influenced by user authorization and permissions. For example, Administrators can see all users and projects assigned to those users in AdminStudio. In the case of NT Groups, Administrators can see individual members of those groups in the **Projects** tab. Further, Administrators can assign projects to users when running the **New Workflow Project Wizard**.

An example of how permissions affect workflows and projects is the display of the **Workflow Templates** tab, which requires the **View Workflow Tab** permission. Likewise, only users with the **Create Project** permission can create projects. Even if you have permission to view and create workflows, you can only associate tools which you are permitted to use with tasks you create. If you are executing projects, you can only launch tools you have permissions to use, regardless of whether they are associated with a task in the workflow.

For more information, see [Managing Roles and Permissions](#).

Frequently Asked Questions

The following is a list of questions frequently asked by AdminStudio users, including a link to the appropriate help topic.

General & Workflow

- How do I add a new tool to the Tools Gallery? See [Adding New Tools to the Tools Gallery](#).
- How do I add a command line configuration to a tool? See [Adding Command-Line Configurations for an Existing Tool](#).
- How do I specify a default Application Catalog? See [Specifying a Default AdminStudio Application Catalog](#).

Application Isolation Wizard

- How do I isolate applications? See [Isolating Applications Using Application Isolation Wizard](#).
- How do I modify the default isolation recommendations? See [Modifying the Default Isolation Recommendations](#).

ConflictSolver

- What types of conflicts can ConflictSolver detect? See [Identifying and Resolving Application Conflicts Using ConflictSolver](#).
- How do I change which conflicts are checked? See [Changing Default Conflict Types Checked](#).
- How do I validate a package? See [Validating Packages](#).
- How do I identify conflicts? See [Checking for Conflicts Using the Conflict Wizard](#).
- How do I automatically resolve conflicts? See [Automatically Resolving Conflicts](#).

Tuner

- What should I do if MSI prevalidation fails? See [Handling Invalid Windows Installer Packages](#).
- How do I prevent a feature from displaying during custom installation? See [Changing a Feature's Visibility](#).
- When should I use the Dialogs View instead of MSI command-line options? See [Dialogs View vs. Command-Line Options](#).
- How do I create a setup.exe file for my package and transform? [Creating a Setup.exe File for the Package and Transform](#).
- When do I use Tuner vs. InstallShield Editor? See [Customizing Installations Using Tuner](#).

Repackager

- Why do people use a Repackager? See [Repackaging Legacy Installations Using the Repackaging Wizard](#).
- Why is a “clean” system important for repackaging? See [About Repackaging on Clean Systems](#).
- Should I repackage a Windows Installer (.msi) setup? See [Repackaging Wizard Best Practices](#).

- How do I repackage a non-Windows Installer setup? [Repackaging Legacy Installations Using the Repackaging Wizard](#) and [Converting Legacy Installations Using the Repackager Interface](#).
- How can I speed up repackaging? See [Repackaging Wizard Best Practices](#).
- What can I do with a repackaged setup? See [Repackaging Legacy Installations Using the Repackaging Wizard](#) and [Converting Legacy Installations Using the Repackager Interface](#).
- Where does Repackager store my repackaged files and the MSI packages it builds? See [Set Target Project Information and Capture Settings Panel](#).
- How do I identify and fix WinINSTALL conversion problems? See [Troubleshooting Guidelines for WinINSTALL Conversion](#).
- How do I identify and fix SMS conversion problems? See [Troubleshooting Guidelines for SMS Conversion](#).
- What do I do if I receive a `ISDEV: fatal error -5023: Error building table file` error while using Repackager? See [Resolving an "Error Building Table File" Error](#).

Distribution Wizard

- How do I create an administrative installation? See [Creating Administrative Installations for Packages](#).

OS Snapshot Wizard

- Why do I need an OS Snapshot? See [Taking OS Snapshots](#).

InstallShield Editor

Answers to common questions regarding InstallShield Editor can be found under **Frequently Asked Questions** in the InstallShield Editor Help Library.

AdminStudio Interface Reference

The AdminStudio interface reference is organized into the following areas:

- [AdminStudio Start Page](#)
- [Assistant Tab](#)
- [Tools Tab](#)
- [Workflow Templates Tab](#)
- [Projects Tab](#)
- [Enterprise Server Tab](#)
- [AdminStudio Menus and Toolbar](#)
- [Dialog Boxes](#)
- [Wizards](#)

AdminStudio Start Page

The AdminStudio Start Page provides quick access to product information, to recently opened files, and to InstallShield resources.

AdminStudio Views

The AdminStudio interface is organized into the following tabs, which appear across the top of the Start Page:

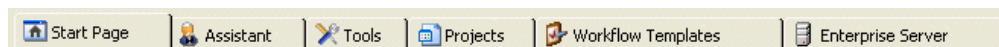


Figure 3-3: AdminStudio Interface Tabs

Click on these tabs to access the following AdminStudio views:

- **Start Page**—Initial view of AdminStudio.
- **Assistant**—The Process Assistant, a procedure-based alternative interface to the standard AdminStudio interface, is accessible on the [Assistant Tab](#).
- **Tools**—The [Tools Tab](#) includes the Tools Gallery and information on the selected tool.
- **Workflow Templates**—On the [Workflow Templates Tab](#), you can create and modify workflows, which are the basis for all projects in AdminStudio.
- **Projects**—On the [Projects Tab](#), you can create, execute, and delete projects, and access existing projects, which are the procedures followed to accomplish a set goal.
- **Enterprise Server**—On the [Enterprise Server Tab](#), you can use Job Manager to automate tasks and Report Center to generate reports. You can also manage Users, Roles, and Permissions.

Tools Gallery

You can launch each tool in the AdminStudio suite by clicking on the appropriate icon in the Tools Gallery.

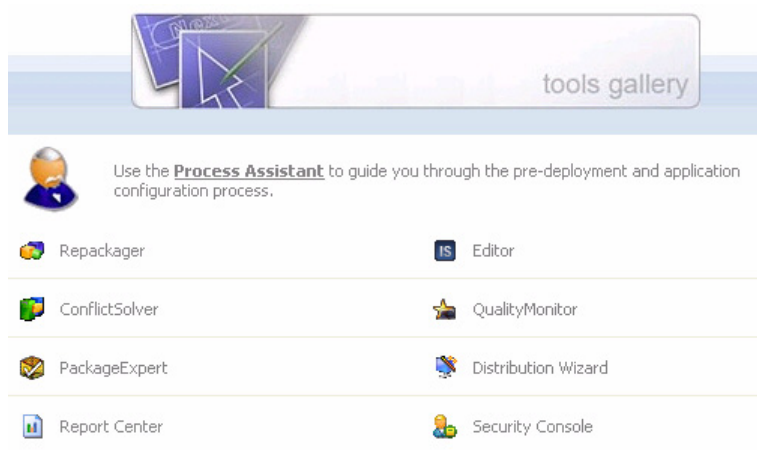


Figure 3-4: Tools Gallery on the AdminStudio Start Page



Note • Individual AdminStudio applications can also be launched by double-clicking on the tool in the Tools Gallery on the **Tools Tab**. See [Tools Tab](#).

Application Tasks and Help Topics

Click on an Application Task to quickly launch the Process Assistant task for packaging a setup, customizing a setup, identifying and resolving application conflicts, ensuring package quality, or preparing packages for distribution. Frequently accessed Help Topics are also listed. To access the entire AdminStudio Help Library from the Start Page, click the Help Library link in the Resources section or click the Help button in the toolbar.

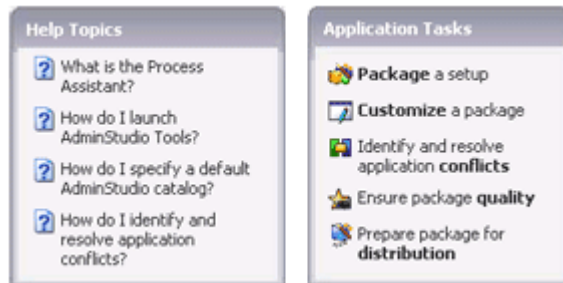


Figure 3-5: Help Topics and Application Tasks Areas

(Recently Opened Files)

The section in the middle of the Start Page lists your most recently accessed files, the task you were performing with that file, and the dates on which they were last modified. Click on a file to quickly launch the Process Assistant task and continue processing that file.

Resources

The Resources section contains a number of links to connect you to helpful InstallShield information and InstallShield's responsive customer support.

Contact Us

To provide feedback about InstallShield or join our Customer Experience Improvement Program, click one of the links listed here.

Assistant Tab

The **Assistant Tab** is the Welcome page to the AdminStudio **Process Assistant**, a procedure-based alternative interface to the standard AdminStudio interface. It guides you through the process of preparing a new application for enterprise use including packaging and feature customization, conflict resolution, QA testing, and deployment hand-off.

The **ConflictSolver Process Assistant**, which is integrated into the main Process Assistant but which can also be launched separately, guides you through the process of identifying and resolving conflicts between Windows Installer Packages.



Note • For more information, see [Using the Process Assistant](#),

Tools Tab

The Tools tab, which is accessed from the AdminStudio **Start Page**, includes the Tools Gallery and the Content Pane.

Tools Gallery

You can launch each tool in the AdminStudio suite by double-clicking on the appropriate icon in the Tools Gallery. Individual AdminStudio applications can also be launched by clicking on the tool in the Tools Gallery on the **Start Page**:

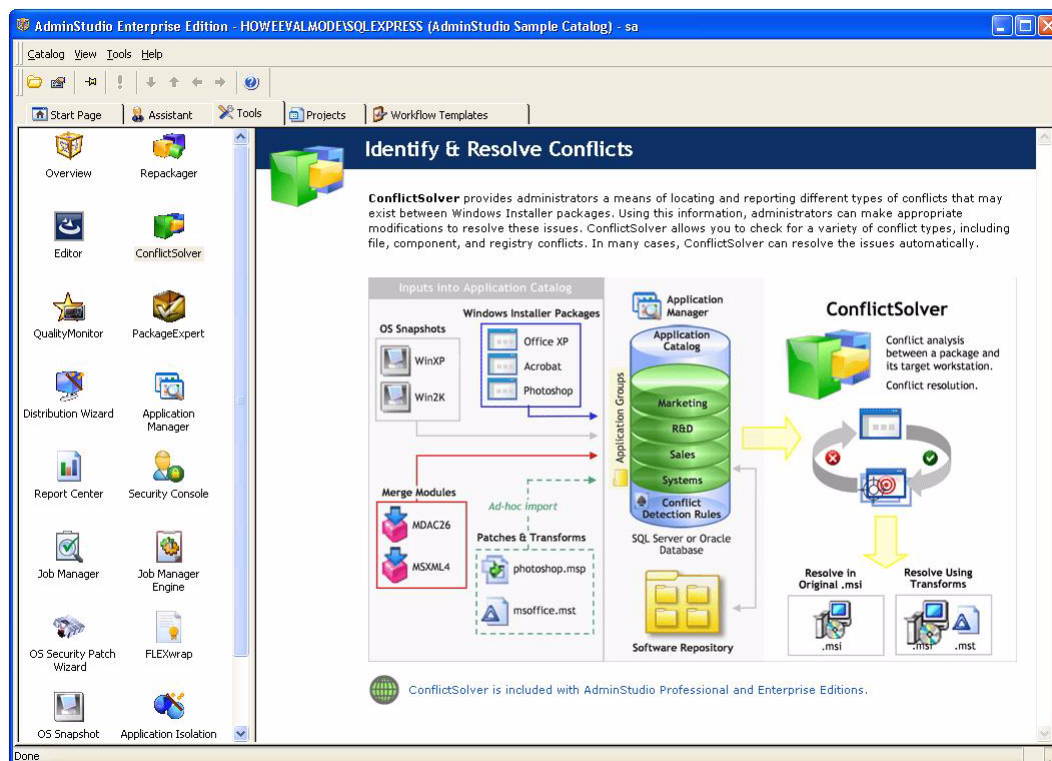


Figure 3-6: Tools Gallery on the Tools Tab



Note • If a core AdminStudio tool is not in the Tools Gallery or available from the menu, it may mean an administrator has restricted tool access based on user permissions.

By right-clicking in the Tools Gallery, you can launch the **Add Tool Wizard**, from which you can add new tools into the gallery and make them available for task assignments in workflows and projects.

Content Pane

The first icon in the Tools Gallery, Overview, is selected by default. When selected, the Content Pane lists a description of each of the AdminStudio tools. If you single-click on an AdminStudio tool icon in the Tools Gallery on the **Tools** tab, additional information about the selected tool is displayed in the Content Pane.


Workflow Templates Tab

Workflows, which can be created and modified from the Workflow Templates tab, are the basis for all projects in AdminStudio. These workflows consist of defined tasks, with which instructions (in the form of HTML files) and tools can be associated. Users can then create projects based on these workflows, and execute them—following the specific steps defined in the workflow. This allows you to create specific, repeatable procedures to accomplish your application migration goals.

Tasks

Once you create a workflow, you can add tasks to it. Tasks are discrete steps in your overall process. Each task has the following configurable options:

Table 3-3 • Task Options

Option	Description
Tool	<p>If needed, you can pick a tool to associate with the task. When a user runs the workflow, the tool can be launched from the workflow step. By default, the AdminStudio tools are included in this list. If you have added tools to the Tools Gallery, they also appear in this list. If you want to add a tool directly from the Workflow Templates tab, select the <New Tool> option to display the Add New Tool dialog box. This adds the tool to the Tools Gallery and makes it available for the current task.</p>  <p>Note • If, after adding a new tool for a task that is not included in the Tools Gallery, you assign a different tool or no tool to the task, the tool you added will no longer be available. To avoid this, when possible, add tools to the Tools Gallery</p>
Tool Configuration	<p>This list contains all available command-line configurations for the selected tool. If you do not need a configuration, select <None>. Click Configure to add new configurations to the tool, which you can then select from this list.</p>
Help File	<p>You can associate a help file (in HTML format) with the task to provide instructions for performing the task. Enter the path and help file in this field, or use the Browse button to navigate to it. If you have yet to create an HTML page, click the Edit HTML button to the right of the Browse button to open a default page in an HTML editor.</p>
Notes	<p>Add any notes you want associated with this task. This field can only hold 255 characters, so additional information should be added to your help file.</p>

Projects Tab

Purpose

From the Projects tab, you can create, execute, and delete projects. You can also access existing projects. The drop-down filter above the Project pane allows you to view all available projects, or only a specific project.

Each project must be based on an existing workflow. In this way, projects are similar to photocopies from a master instruction sheet—all planning and design of the procedure is done to the workflow. The project is a copy of that workflow, and multiple projects can be based on the same workflow if you are performing the same procedure.

Integration with AdminStudio Workflow Manager

AdminStudio Workflow Manager is a Web-based application that manages the application lifecycle, incorporating standards (data) and methodologies (process). AdminStudio Workflows and Workflow Manager Workflows can be integrated, so that an AdminStudio Project can be a Workflow Phase in a Workflow Manager Workflow.

When an AdminStudio Project is linked to a Workflow Manager Workflow, please note the following indications on the Projects tab:

- When an integrated AdminStudio project is selected on the Projects tab, the name of its associated Workflow Manager Application is displayed in the Project Properties.
- When an AdminStudio Project is linked to a Workflow Manager Workflow and the workstation is not currently connected to the Workflow Manager Server, the following icon appears in the bottom right of the Projects tab view:



Enterprise Server Tab



Edition • The **Enterprise Server** tab is available in AdminStudio Enterprise Edition.

AdminStudio Enterprise Server is a security console and set of Web tools that are closely integrated with AdminStudio. The Enterprise Server tools include the Security Console, Job Manager, Report Center, Configuration Manager Web Console, and Workflow Manager. For more information, see the following topics:

- [Generating and Viewing Reports in Report Center](#)
- [Automating Tasks Using Job Manager](#)
- [Distributing Packages Using Configuration Manager Web Console](#)
- [Managing Roles and Permissions](#)
- [Managing Users, Directory Services, and User Logins](#)

Workflow Manager Tab

AdminStudio Workflow Manager is a Web-based application that manages the application lifecycle, incorporating standards (data) and methodologies (process). AdminStudio Workflows and Workflow Manager Workflows can be integrated, so that an AdminStudio Project can be a Workflow Phase in a Workflow Manager Workflow.

If you have purchased AdminStudio Enterprise Edition and Workflow Manager, the Workflow Manager icon will be displayed in the AdminStudio Tools Gallery on the AdminStudio Tools tab and will be available as a tab on the AdminStudio Enterprise Server interface.



Note • For more information on Workflow Manager, see **Using Workflow Manager to Manage Enterprise Software Packaging** in the AdminStudio Enterprise Server Help Library.

AdminStudio Menus and Toolbar

The following commands and toolbar buttons are available in the AdminStudio interface.

Table 3-4 • AdminStudio Menus and Toolbar



Menu	Command	Shortcut	Button	Description
Catalog	Connect	Ctrl+O		Displays the Connect Application Catalog Dialog Box , where you can open an existing SQL Server Application Catalog or the AdminStudio Enterprise Server Application Catalog.
Catalog	Create	Ctrl+N		Displays the Application Catalog Wizard , where you can create a new SQL Server Application Catalog database.
Catalog	Disconnect	Ctrl+D		Closes the currently open Application Catalog.
Catalog	Properties			Opens the Application Catalog Properties Dialog Box where you can configure the Software Repository.
Catalog	Change AES Password			Change the password of the current user to log in to the AdminStudio Enterprise Server.
Catalog	Logout			Log out of the AdminStudio Enterprise Server.

Table 3-4 • AdminStudio Menus and Toolbar (cont.)







Menu	Command	Shortcut	Button	Description
Catalog	Replication	Ctrl+R		Use to access the SQL Server Application Catalog Replication feature. These menu items are selectively enabled depending on whether the current user is connected to an SQL Server database and has been assigned sysadmin privileges in SQL Enterprise Manager on that database server. If the open database is a Subscriber database, the Subscription Manager option will be enabled, and if it is a Publisher database, the Publication Manager option will be enabled.
Catalog	Exit	Alt+C+X		Exits AdminStudio and returns you to the Windows desktop.
View	Toolbar	Alt+V+T		Toggles the Toolbar.
View	Status Bar	Alt+V+S		Toggles the Status Bar.
View	Always On Top	Alt+V+A		When checked, the AdminStudio Interface remains on top of all other windows.
View	Start Page			Select to open the AdminStudio Start Page.
View	Process Assistant			Select to open the Welcome Page of the Process Assistant.
View	Tools			Select to open the Tools tab, which includes the Tools Gallery and information on the selected tool.
View	Projects			Select to open the Projects tab. Projects, which are based on existing workflows, are the procedures followed to accomplish a set goal.
View	Workflow			Select to open the Workflow Templates tab, where you can create and modify workflows. Workflows serve as templates upon which projects are based.
Tools	Check for Updates	Alt+T+U		Start the Update Service to determine if there any updates or messages available for AdminStudio.
Tools	Options	Alt+T+O		Displays the Options dialog box, from which you can configure the location of shared resources and the frequency AdminStudio checks for updates.

Table 3-4 • AdminStudio Menus and Toolbar (cont.)

Menu	Command	Shortcut	Button	Description
Help	Contents	Alt+H+C		Launches the online Help Library and displays the Contents tab.
Help	Index	Alt+H+I		Launches the online Help Library and displays the Index tab.
Help	Search	Alt+H+S		Launches the online Help Library and displays the Search tab.
Help	Support Central	Alt+H+U		Connects to the AdminStudio Support Web site.
Help	Web Community	Alt+H+M		Connects to the AdminStudio Web Community.
Help	ReadMe	Alt+H+R		Displays the AdminStudio Release Notes file.
Help	Feedback	Alt+H+F		Connects to an online form, through which you can provide feedback about AdminStudio.
Help	AdminStudio on the Web	Alt+H+W		Connects to the AdminStudio Web site.
Help	About AdminStudio	Alt+H+A		Displays the About dialog box with version information for AdminStudio.
Context Menu	New Workflow			Create a new Workflow.
Context Menu	New Task			Create a new Task.
Context Menu	Rename			Rename selected Workflow or Task.
Context Menu	Delete			Delete selected Workflow or Task.
Context Menu	New Project		Ctrl-P	Create a new Project.
Context Menu	Del Project			Delete selected Project.

Table 3-4 • AdminStudio Menus and Toolbar (cont.)

Menu	Command	Shortcut	Button	Description
Context Menu	Run Task			Runs the tool associated with the selected task in the project.
Context Menu	Move Up			Moves the selected task up in the task order.
Context Menu	Move Down			Moves the selected task down in the task order.
Context Menu	Move Left			Moves the selected task left in the task order.
Context Menu	Move Right			Moves the selected task right in the task order.

Dialog Boxes

The following dialog boxes can be opened from the AdminStudio Interface:

- [About AdminStudio Dialog Box](#)
- [Add New Tool Dialog Box](#)
- [Command Line Properties Dialog Box](#)
- [Options Dialog Box](#)
- [Tool Properties Dialog Box](#)

About AdminStudio Dialog Box


The **About AdminStudio** dialog box can be opened by selecting About AdminStudio from the **Help** menu. This dialog box displays information about the product, including the full version number (essential if you need technical support). If you have not registered AdminStudio, click the **Register** button to connect to the InstallShield Web site to begin the Product Registration process. Registering your product offers you expert technical support, new product announcements and special offers, plus notification of product upgrades.

Add New Tool Dialog Box

The Add New Tool dialog box is displayed if you select <New Tool> from the Tool list while designing a workflow. This dialog box allows you to provide information about a tool you want accessible from AdminStudio, particularly to use in workflow tasks.

This dialog box contains the following options:

Table 3-5 • Add New Tool Dialog Box Options

Option	Description
Target	Enter the location of the tool or Browse to the application or file you want added to the tools pane for use in AdminStudio.
Name in Tools Gallery	This description is used as the display name for the application in the Tools Gallery.
Command Line Arguments	Enter any command-line arguments for the tool. Because you may have different uses for applications, you can add the same application multiple times to the tools pane, with each instance using different command line arguments.
Working Directory	If this tool requires a working directory, enter it here or click Browse to locate it.
Comments	Enter any comments about this tool in this field.
HTML Explanation File	Enter the location and name of an HTML file you want displayed when you single-click on the tool in the tools pane. Alternatively, click Browse and navigate to it. If you have yet to create one, click the Edit HTML button () below the field to open a default page in an HTML editor:
Add to Tools Gallery Check Box	When this box is checked, the tool will be added to the Tools Gallery. If unchecked, it is only available for the task where it was added.

Command Line Properties Dialog Box

The Command Line Properties dialog box is displayed when you create or edit a command-line configuration for a tool.

The Command Line Properties dialog box has two configurable options:

Table 3-6 • Command Line Properties Dialog Box Options

Option	Description
Description	Provide a description for the configuration. This assists you in differentiating similar command-line options.
Command Line	<p>Provide the actual command-line parameters for the tool. The arrow to the left allows you to select one of the following variables to include in the command-line:</p> <ul style="list-style-type: none"> • InstallLocation—The location where AdminStudio is installed. • DevLocation—The location where InstallShield Editor is installed. • SharedPoint—The AdminStudio shared directory. • SourcePackage—The name and location of the source package. • TargetDir—The directory where output from the selected project is stored. • TargetFileName—The name of the output file. • ProjectName—The name of the current project.

Options Dialog Box

From the Options dialog box, you can configure settings including application catalog settings, shared locations settings, and the frequency AdminStudio checks for updates. The dialog box consists of the following tabs:

- [Application Catalog Tab](#)
- [Locations Tab](#)
- [Updates Tab](#)
- [Virtual Packages Tab](#)
- [Quality Tab: Customer Experience Improvement Program](#)

Application Catalog Tab

From this tab, you can configure the following options:


Table 3-7 • Options Dialog Box/Application Catalog Tab Options

Option	Description
Shared AdminStudio Application Catalog	This field displays the name and location of the shared AdminStudio Application Catalog. Click the Change button to select another Application Catalog using the Select Application Catalog Dialog Box .
AdminStudio Enterprise Server URL	Enter the URL for the AdminStudio Enterprise Server, available in the AdminStudio Enterprise Edition.
Make this the default shared AES URL	If you have more than one AdminStudio Enterprise Servers, select this option to make the URL in the AdminStudio Enterprise Server URL field the default.
Allow automatic authentication	Select this option to allow automatic authentication by AdminStudio Enterprise Server.
Install AES	If AdminStudio Enterprise Server is not installed and you want to install it on this machine, click this button to launch the installation.

Locations Tab

From this tab, you can configure the following options:

Table 3-8 • Options Dialog Box/Locations Tab Options

Option	Description
AdminStudio Shared Location	<p>Enter or browse to the shared location for AdminStudio. This location will contain shared information for repackaging and conflict identification. To maintain consistency when creating workflows, it is recommended that you set this shared location the same for each AdminStudio seat.</p>  <p>Tip • The AdminStudio Shared Location is defined during installation and normally does not need to be changed. It is usually assigned to a network folder, preferably a UNC path.</p>
Task Help Page Location	Enter or browse to the directory where you want to store all HTML pages that serve as workflow task instructions.
Configuration Manager Web Console URL	Enter the URL for the Configuration Manager Web Console Web site.

Updates Tab

You can configure the following options from this tab:

Table 3-9 • Options Dialog Box/Updates Tab Options

Option	Description
Check for software updates	Specify how often you want AdminStudio to check for updates. Your options are: <ul style="list-style-type: none"> • Never • Once every 15 days • Once every 30 days (default) • Once every 60 days

Virtual Packages Tab

On the **Virtual Packages** tab, you can control which of the available virtual package formats are displayed in the AdminStudio toolset—Microsoft App-V, VMware ThinApp, and/or Citrix XenApp—and you can choose whether to include the App-V Application Launcher with App-V applications.

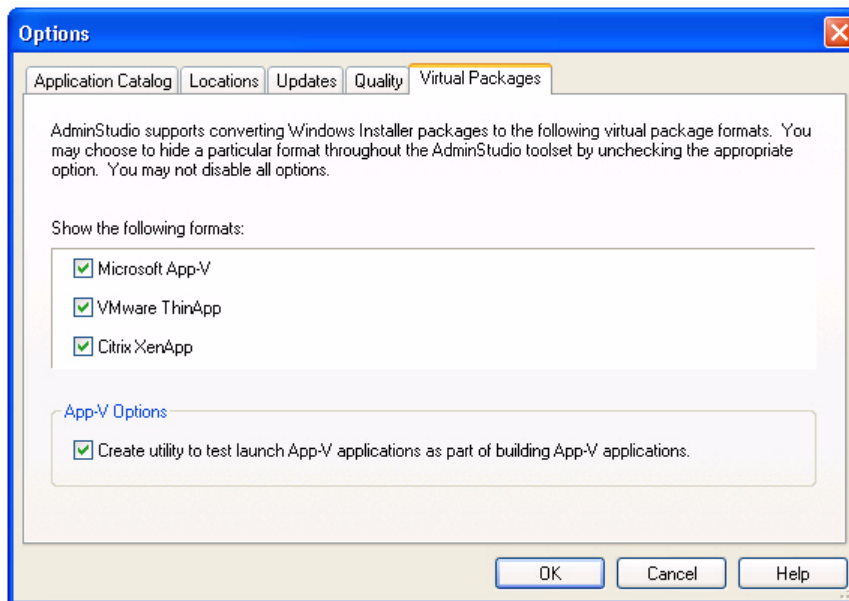



Figure 3-7: Virtual Packages Tab of the Options Dialog Box

The **Virtual Packages** tab of the **Options** dialog box has the following options:

Table 3-10 • Virtual Packages Tab of the Options Dialog Box

Option	Description
Microsoft App-V	Select to display all Microsoft App-V virtual package format options throughout AdminStudio and InstallShield Editor. If this check box is not selected, the options for the virtual package format will be hidden.
VMware ThinApp	Select to display all VMware ThinApp virtual package format options throughout AdminStudio and InstallShield Editor. If this check box is not selected, the options for the virtual package format will be hidden.
Citrix XenApp	Select to display all Citrix XenApp virtual package format options throughout AdminStudio and InstallShield Editor. If this check box is not selected, the options for the virtual package format will be hidden.
Create utility to test launch App-V applications as part of building App-V applications.	<p>Select this option to include the App-V Application Launcher with each App-V application that you build. You can use the App-V Application Launcher to locally test a newly built App-V application before moving it to a deployment server. For more information, see Testing App-V Applications Using the App-V Application Launcher.</p>  <p>Note • This option is only visible when the Microsoft App-V format is selected.</p>



Important • You may not disable all options.

Quality Tab: Customer Experience Improvement Program

The Customer Experience Improvement Program (CEIP) helps AdminStudio improve the quality, reliability and performance of our software and services.

If you choose to participate in the Customer Experience Improvement Program, we will collect anonymous information about how you use AdminStudio. This information helps us identify trends and usage patterns.

All information collected is anonymous, and this data collection will not affect the performance of AdminStudio tools. You will never be prompted to complete a survey, and no one from our company will contact you. You can continue working with AdminStudio without interruption.

Your membership status in the Customer Experience Improvement Program is specified on the **Quality** tab of the AdminStudio **Options** dialog box. If you initially select to participate but later you change your mind, you can opt-out of this program by changing the selection on the **Quality** tab.

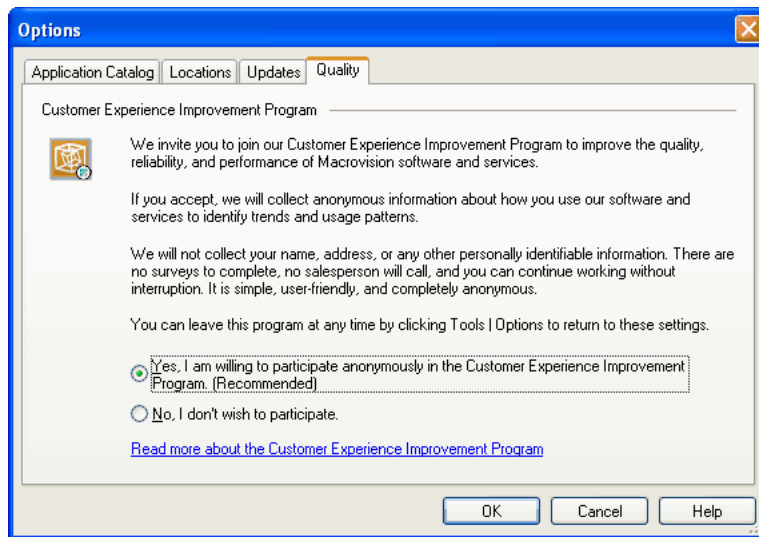


Figure 3-8: Quality Tab of the AdminStudio Options Dialog Box

Participation in the Customer Experience Improvement Program is not mandatory, but your input is appreciated.

For more information on the Customer Experience Improvement Program, visit the AdminStudio Web Site.

Tool Properties Dialog Box

The Tool Properties dialog box is displayed when you right-click on a tool in the Tools gallery and select Properties. This dialog box contains the following tabs:

- [Properties Tab](#)
- [Configuration Tab](#)

Properties Tab


The Properties tab of the Tool Properties dialog box contains information about the tool, including the name and location of the executable, the name of the tool as it appears in the Tools gallery, and the help file associated with it (if any).

The following options can be configured:

Table 3-11 • Tool Properties Dialog Box/Properties Tab Options

Option	Description
Target	Enter the location of this tool's executable. Alternatively, click Browse and navigate to it.
Name in Tools Gallery	Provide a name for the tool as it will appear in the Tools gallery.

Table 3-11 • Tool Properties Dialog Box/Properties Tab Options (cont.)

Option	Description
Command Line Arguments	Enter any default command line arguments for this tool.
Working Directory	If this tool requires a working directory, enter it here or click Browse to locate it.
Comments	Provide any comments about this tool.
HTML Explanation File	Enter the location and name of an HTML file you want displayed when you single-click on the tool in the tools pane. Alternatively, click Browse and navigate to it. If you have yet to create one, click the Edit HTML button below the field (shown below) to open a default page in an HTML editor: 

Configuration Tab

From the **Configuration** tab of the Tool Properties dialog box, you can **Add**, **Modify**, or **Delete** command-line configurations associated with the tool. Each tool can have multiple configurations associated with it, for different uses.

Table 3-12 • Tool Properties Dialog Box/Configuration Tab Options

Option	Description
Command Line List	Listing of all command lines defined for this tool.
Add	Click to open the Command Line Properties Dialog Box , where you can add a new command line. In the Description field, you provide a description for the configuration. This assists you in differentiating similar command-line options. In the Command Line field, you provide the actual command-line parameters for the tool. The arrow to the left allows you to select one of the following variables to include in the command-line: <ul style="list-style-type: none">● InstallLocation—The location where AdminStudio is installed.● DevLocation—The location where InstallShield Editor is installed.● SharedPoint—The AdminStudio shared directory.● SourcePackage—The name and location of the source package.● TargetDir—The directory where output from the selected project is stored.● TargetFileName—The name of the output file.● ProjectName—The name of the current project.
Modify	Click to open the Command Line Properties Dialog Box , where you can modify the selected command line.

Table 3-12 • Tool Properties Dialog Box/Configuration Tab Options (cont.)

Option	Description
Delete	Click to delete the selected command line.



Note • Most AdminStudio Tools have one or more command-line configurations already defined. Although you can change or remove these configurations, there is no way to automatically reset them to their default values.

Ways to Assign a Command Line to a Tool

There are three ways a user can assign a command line to a tool:

- **Tool Properties Panel of Add Tool Wizard**—When you add a tool, you can assign a command line on the **Tool Properties Panel** of the **Add Tool Wizard**.
- **Properties Tab of the Tool Properties Dialog Box**—When you view the tool's properties, you can assign a command line on the **Properties Tab** of the **Tool Properties** dialog box.
- **Configuration Tab of the Tool Properties Dialog Box**—Using the **Configuration Tab** of the **Tool Properties** dialog box, you can create *multiple* command lines and can use AdminStudio variables in these command lines. Then, when you go to the Workflow tab and create a new Workflow, you can associate a Tool with a task and also select which command line configuration they want to use. Once you have done that, you can go to Project tab and create a new Project. When you create a new Project, you will have to specify the Source Package and the target directory and file name. Once the Project is created, when you execute the task, AdminStudio will execute the command line configuration you previously selected by replacing the AdminStudio variables in the command line.



Note • A command Line entered by the user in **Properties Tab** of the **Tool Properties** dialog box play no role in the Workflow and Project tab. This command line is used only when you run the tool from the Tools Tab.

Wizards

The AdminStudio interface includes the following Wizards:

- [Add Tool Wizard](#)
- [New Workflow Project Wizard](#)

Add Tool Wizard

The Add Tool Wizard allows you to add new tools that appear in the AdminStudio Tools gallery. You can specify the tool's executable, provide command-line options for the tool, and provide a link to information about the tool.

The Add Tool Wizard includes the following panels:

- [Welcome Panel](#)
- [Tool Properties Panel](#)
- [Command-Line Configurations Panel](#)

Welcome Panel

The Add Tool Wizard allows you to add new tools that appear in the AdminStudio Tools gallery. You can specify the tool's executable, provide command-line options for the tool, and provide a link to information about the tool.

Click **Next** to proceed to the **Tool Properties Panel**.

Tool Properties Panel


From the Tool Properties panel of the Add Tool Wizard, you can enter information about the tool you are adding to the gallery.

You can configure the following options:

Table 3-13 • Add Tool Wizard/Tool Properties Panel Options

Option	Description
Target	Enter the location and name of this tool's executable. Alternately, click Browse and navigate to it.
Name in Tools Gallery	Provide a name for the tool as it will appear in the Tools Gallery.
Command Line Arguments	Enter any default command line arguments for this tool.
Working Directory	If this tool requires a working directory, enter it here or click Browse to locate it.

Table 3-13 • Add Tool Wizard/Tool Properties Panel Options (cont.)

Option	Description
Comments	Provide any comments about this tool.
HTML Explanation File	<p>Enter the location and name of an HTML file you want displayed when you single-click on the tool in the tools pane. Alternatively, click Browse and navigate to it.</p> <p>If you have yet to create one, click the Edit HTML button below the field (shown below) to open a default page in an HTML editor:</p> 

Command-Line Configurations Panel

From this panel, you can add command-line configurations for the tool. Each tool can have multiple command-line configurations for different tool uses.

Click **Add** to add a new option from the **Command-Line Properties** dialog box. You can also click **Modify** to edit the selected command-line option, or **Delete** to remove the selected option.

Click **Finish** to add the tool to the Tools Gallery; click **Back** to return to the **Tool Properties Panel**.

New Workflow Project Wizard

The New Workflow Project Wizard assists you in creating a new project based on an existing workflow. The values you specify in this Wizard are stored in variables that can be accessed from tools, allowing for greater interoperability in AdminStudio. The New Workflow Project Wizard includes the following panels:

- [Welcome Panel](#)
- [Workflow Selection Panel](#)
- [Source Package Panel](#)
- [Target Directory and File Name Panel](#)

Welcome Panel

The New Workflow Project Wizard assists you in creating a new project based on an existing workflow. The values you specify in this Wizard are stored in variables that can be accessed from tools, allowing for greater interoperability in AdminStudio.

Workflow Selection Panel

From the Workflow Selection panel, you can specify the workflow on which you want to base this project. Available workflows appear in the workflows window.

Select the workflow you want to use, and provide the name for the new project (which is stored in the `ProjectName` predefined variable).

Source Package Panel

From the Source Package panel, you can specify the name and location of the source package used in this workflow. If you are creating a repackaging project, this is usually an executable, such as `Setup.exe`. The source package name and location is stored in the predefined variable `SourcePackage`.



Note • You must specify the name and location of a source package if it is required in the workflow (by using the `SourcePackage` variable). For example, if the workflow specifies to launch a package with certain command-line parameters, AdminStudio needs to know what package to launch.

Source packages can also be non-setup files. For example, if you are creating a simple workflow that involves editing a Notepad file, the source package may be a `.txt` file.

Target Directory and File Name Panel

From the **Target Directory and File Name** panel, specify the **Target Directory** and **Target File Name** used in this project.

All output files (such as an INC file from Repackager or an MST file from Tuner) associated with the project will be stored in the Target Directory, and the value for this directory can include a predefined variable such as `SharedPoint`. The Target File Name is the name used for all files created by project tasks in this project, with the appropriate extension appended to it depending on the file type. The Target Directory is stored in the predefined variable `TargetDir` and the output file name is stored in the variable `TargetFileName`.



Note • If any workflow tasks use the `TargetDir` or `TargetFileName` variables, you must specify the target directory and package name. For example, if the workflow specifies to save a task's output, AdminStudio needs to know where to save it and what to call it.

Click **Finish** to close the Wizard.

Log Files

AdminStudio tools generate the following log files:

Table 3-14 • AdminStudio Log Files

Log File	Location
AdminStudio.log	C:\Program Files\AdminStudio\9.5
predeploytest.log	C:\Program Files\AdminStudio\9.5\PreDeployTest\Client
licwrapserver.log	C:\Program Files\AdminStudio\9.5\FLEXWrap\Server
debug.log	C:\Program Files\AdminStudio\9.5\FLEXWrap\Server
distributer.log	C:\Program Files\AdminStudio\9.5\Common
islc.log	C:\Program Files\AdminStudio\9.5\Repackager

Part 2

Managing Users, Roles, Permissions, Application Catalogs

This part of the AdminStudio 9.5 User Guide includes the following chapters:

- [Managing Users, Directory Services, and User Logins](#)
- [Managing Roles and Permissions](#)
- [Managing Application Catalog Databases](#)

Managing Users, Directory Services, and User Logins

You can create a user account for each person that you want to have access to AdminStudio Enterprise Server, or you can import users or groups of users from a Windows Active Directory or Novell eDirectory Directory Service.

You can also configure various login methods to best suit your needs: AdminStudio account login, domain account login, single sign-on login, and guest account login.

Information is presented in the following main sections:

- [Managing Users and Groups](#)
- [Managing Directory Services Configurations](#)
- [Managing User Logins](#)

Managing Users and Groups

There are several methods for creating new users in AdminStudio Enterprise Server:

- **Create a user account**—Create a user account for each person that you want to have access to AdminStudio Enterprise Server. See [Creating a New User Account](#).
- **Import Directory Service users**—Set up a Windows Active Directory or Novell eDirectory Directory Service Connection and import users from that Directory Service. See [Importing Directory Services Users and Groups](#).
- **Import Directory Service group**—Set up a Windows Active Directory or Novell eDirectory Directory Service Connection and import a group from that Directory Service. This allows you to provide dynamic access to all of the members of that group as the members of that group changes. For more information, see [Importing Directory Services Users and Groups](#).



Note • For more information on the methods for logging into AdminStudio Enterprise Server, and the how authentication is performed, see [Managing User Logins](#).

Creating a New User Account

To create a new user account, perform the following steps.



Task:

To create a new user account:

1. Open AdminStudio Enterprise Server and open the **Administration** tab.
2. Open the **Users & Groups** tab. The **User Administration** page opens.
3. Click **Add**. The **User Details** view opens.
4. Select the new user's **Company** from the list.
5. Enter a **User Name** and a **Password**, with the **User Name** in the format of:

`username@companyname.com`
6. Optionally, enter the user's **Email address**.
7. Optionally, enter a geographic **Location** that you can use to group users together, such as **New York Office** or **Midwest Region**, etc. Assigning a geographical location to a user enables you to display this information in reports.
8. All of the **Roles** that have been defined for the selected Company are listed. Select the Roles that you want to assign to this user.



Note • The Roles that a user is assigned to determine what portions of AdminStudio Enterprise Server and Workflow Manager functionality they will have access to. Therefore, you must assign at least one Role to a user.

9. Click **Update** to save your entries and close the **User Details** view, or click **Update and Add New** to remain in the **User Details** view so you can add another user.

Importing Directory Services Users and Groups

If you have defined a Directory Service, as described in [Adding a Directory Service Connection](#), you can choose to add Users or Groups from that Directory Service into AdminStudio Enterprise Server.

When a user is added from a Directory Service into AdminStudio Enterprise Server, only information that uniquely identifies the user in the Directory Service is stored in AdminStudio Enterprise Server. Information such as telephone number and email address will always be queried at run time, so that the most current information is obtained.

AdminStudio Enterprise Server supports Windows Active Directory and Novell eDirectory directory services.



Note • For more information on AdminStudio Enterprise Server support for Directory Services, see [Managing Directory Services Connections](#) and [Managing User Logins](#).



Task: *To import users or groups from a Directory Service into AdminStudio Enterprise Server:*

1. Open AdminStudio Enterprise Server and open the **Administration** tab.
2. Open the **Users & Groups** tab. The **User Administration** page opens.
3. Click **Directory Service User/Group Import**. The **Directory Services User/Group Add** view opens.
4. From the **Select a Directory Service** list, select the Directory Service that you want to import users or groups from.
5. Next to **Select Group or User**, select the **User** or **Group** option.
6. Next to **Filter Directory Service** list, select one of the following options:
 - **Show All**—Select this option to include all Users or Groups in the list.
 - **Filter list by**—Select this option, enter text in the box, and click **Find** to filter the list by the specified criteria.

For example, to search for all of the users that start with the letter **P**, use the asterisk wildcard character (*) and enter **P*** in the **Filter list by** box.

7. From the **Select Group or User to Add** list, select the group or user to import to AdminStudio Enterprise Server. The **User Details** view opens, providing a read-only view of the imported user or group's User Name and Password.

If you import a User that is a member of a Group that was previously imported, that User inherits the Roles that are assigned to that Group. You can then assign additional Roles to that User.
8. Click **Update** to save your entries and close the **User Details** view, or click **Update and Add New** to remain in the **User Details** view so you can add another user.

Disabling a User or Group Account

User and Group accounts have a status of **Active** or **Inactive**. If a User or Group has the status of **Inactive**, that User or a member of that Group:

- Cannot log into AdminStudio Enterprise Server.
- Cannot be assigned any work.
- Is not listed on the **User Administration** page.
- Does not get any email notifications.
- Cannot be selected as a **Customer Contact** or **Servicer Contact**.
- Cannot be viewed by Customers.

Deleting vs. Disabling a User Account

If a user is associated in any way with an open or completed Workflow Manager Application Request, you cannot delete that user from the system. In that case, you would instead disable that account by setting its **Status** to **Inactive**.

- **If a user IS NOT associated with an Application Request**, you can delete that user. This means that if you create a user by mistake, before that user has interacted with the application, you can delete it by opening that user on the **User Details** view and clicking **Delete**.
- **If a user IS associated with an Application Request**, However, since AdminStudio Enterprise Server stores the IDs of users on actions and assignments for historical, tracking and reporting purposes, you cannot delete a user that is associated with a Workflow Manager Application Request. In this case, you would set the user's **Status** to **Inactive** on the **User Details** view.



Task: *To disable a user account:*

1. Open AdminStudio Enterprise Server and open the **Administration** tab.
2. Open the **Users & Groups** tab. The **User Administration** page opens.
3. Select the User or Group that you want to disable to open its **User Details** view.
4. Select **Inactive** from the **Status** list and click **Update**.

Deleting a User Account

You can delete a User or Group account from the **User Details** view.



Task: *To delete a user account:*

1. Open AdminStudio Enterprise Server and open the **Administration** tab.
2. Open the **Users & Groups** tab. The **User Administration** page opens.
3. Select the User or Group that you want to delete to open its **User Details** view.
4. Click **Delete**. You are prompted to confirm the deletion.
5. Click **OK**. The deleted User or Group is no longer listed on the **User Administration** page.

Managing Directory Services Configurations

AdminStudio Enterprise Server can be integrated with Windows Active Directory and Novell eDirectory. This enables you to set up automatic login with AdminStudio Enterprise Server based upon Directory Services authentication.

If you import a Group, all members of that group could then login to AdminStudio Enterprise Server without requiring you to import the individual members of the Group. AdminStudio Enterprise Server can then retrieve user attributes, such as email address or telephone number, from the Directory Service dynamically.

- **Directory Service Connections**—Rather than creating a user account for each person who will use AdminStudio Enterprise Server, you can import users from Windows Active Directory or Novell eDirectory directory services. For information on integrating AdminStudio Enterprise Server with Directory Service users and groups, see [Managing Directory Services Connections](#).
- **Directory Service Attributes**—When creating a Workflow Manager Template, you can set a Data Element's **Data Type** to **Directory Service**. This means that when a Directory Service authenticated user performs the Workflow Step that requests that Data Element, information is pulled from the Directory Service to automatically populate the field, such as that user's name, email address, or location. For information on enabling the use of Directory Service Attributes, see [Managing Directory Services Attributes](#).



Note • For more information about Directory Services and Lightweight Directory Access Protocol (LDAP), see [Ldp Overview](#) on the Microsoft TechNet Web site.

Managing Directory Services Connections

Rather than creating a user account for each person who will use AdminStudio Enterprise Server, you can import users from Windows Active Directory or Novell eDirectory directory services. To integrate AdminStudio Enterprise Server with a Directory Service users and group, you set up a **Directory Service Connection**.

This section includes the following topics:

- [Adding a Directory Service Connection](#)
- [Editing an Existing Directory Service Connection](#)
- [Deleting a Directory Service Connection](#)

Adding a Directory Service Connection

Directory Services Connections are used to import users and groups into AdminStudio Enterprise Server, and to authenticate Active Directory or eDirectory users logging into AdminStudio Enterprise Server.



You can choose to have a Directory Service listed in the **Domain** list on the AdminStudio Enterprise Server login page. This enables users in this Directory Service to login to AdminStudio Enterprise Server using their enterprise network credentials.


To add a Directory Service connection, perform the following steps:



Task: *To add a Directory Service connection:*

1. On the **Administration** tab, click **Directory Services**. The **Directory Services** page opens.
2. Click **Manage Directory Services Connections**. The **Directory Services List** page opens.
3. Click **Add**. The **Add Directory Service Connection** view opens.
4. Enter the following information:

Option	Description
Use to Authenticate Users?	<p>Select Yes to include this Directory Service in the Domain list on the AdminStudio Enterprise Server login page. This enables users to login to AdminStudio Enterprise Server using their enterprise network credentials.</p> <p>You can create many Directory Service connections, but only those connections that have this field checked will be listed in the Domain list on the AdminStudio Enterprise Server login page.</p>  <p>Note • In an enterprise, there is usually only one Directory Service that is responsible for authenticating users. However, you can create additional Directory Service connections to import users and groups.</p>
Directory Service Name	Enter a name to identify this Directory Service in the Domain list on the AdminStudio Enterprise Server login page. It is recommended that the Directory Service Name match the Domain Name.
Description	Enter a description of this Directory Service.
Directory Service Type	Identify the type of Directory Service that you are adding a connection to: <ul style="list-style-type: none">• Active Directory• Novell eDirectory
Directory Service Host	Enter the name or IP address of the Server on which this Directory Service is running.
Directory Service Port	<p>Enter the port number of the Server on which this Directory Service is running. By default, AdminStudio Enterprise Server uses Port 389.</p>  <p>Important • The value for the Directory Service Port should be the port on the given server to which AdminStudio Enterprise Server should connect to send LDAP queries.</p>

Option	Description
Base Distinguished Name	<p>Enter the Base Distinguished Name (DN) to identify the root node of this Directory Service.</p> <p>For example, for MyCompany, the Base DN could be:</p> <pre>dc="MyCompany", dc="com"</pre>
Domain Name	<p>Enter the Domain Name of this Directory Service. This Domain Name is used to create the right credentials, along with the Admin User Name. For example, you would enter DomainName\AdminUserName to connect to the specified Directory Service.</p>  <p>Note • To upgrade from AMS 2.5, enter the same Domain Name that was used in AMS 2.5. AdminStudio Enterprise Server will read this Domain Name and then try to authenticate the User using this Directory Service. For single sign-on to work on an upgrade scenario from 2.5 to 4.0, administrators have to configure at least one Directory Service.</p>
Use Secure Socket Layer (SSL)?	Select Yes if this Directory Service is configured to use Secure Socket Layer (SSL).
Connect Anonymously?	<p>Select No if you do not want to permit anonymous connections.</p> <p>If you select Yes to permit anonymous connections, AdminStudio Enterprise Server may not be able to authenticate Directory Service users and may not be able to add Directory Service users/groups into AdminStudio Enterprise Server.</p>
Admin Distinguished Name	<p>Enter the Distinguished Name of a user who has permission to retrieve users/groups information and authenticate any user against this Directory Service. Enter the Distinguished Name in the format of:</p> <pre>domainname\username</pre> <p>such as:</p> <pre>abcinc\jsmith</pre>
Password	Enter the password for the Admin Distinguished Name.
Group Class Name	<p>Enter the object class names for groups for this Directory Service. Default values are:</p> <ul style="list-style-type: none"> • Active Directory: group • eDirectory: groupofnames
Group Name Attribute	Enter an attribute used by this Directory Service to name groups. The default value for both Active Directory and eDirectory is cn.

Option	Description
Group Member Attribute	Enter an attribute used by this Directory Service to define member groups. Default values are: <ul style="list-style-type: none">• Active Directory: member• eDirectory: uniquemember
User Class Name	Enter the object class name used by this Directory Service for users. Default values are: <ul style="list-style-type: none">• Active Directory: user• eDirectory: inetorgperson
User Name Attribute	Enter the attribute used by this Directory Service to identify users. Default values are: <ul style="list-style-type: none">• Active Directory: samaccountname• eDirectory: uid

5. Click **Test Connection** test to see if the settings that you entered can be used to successfully connect to this Directory Service.
6. Click **Update** to save your entries and return to the **Directory Services List** page or click Update and Import (User/Group) to save your entries and open the **Directory Services User/Group Add** view.

Editing an Existing Directory Service Connection

To edit an existing Directory Service connection, perform the following steps:



Task: *To edit an existing Directory Service connection:*

1. On the **Administration** tab, click **Directory Services**. The **Directory Services** page opens.
2. Click **Manage Directory Services Connections**. The **Directory Services List** page opens.
3. Select the Directory Service connection that you want to edit. The **Edit Directory Service Connection** page opens.
4. Edit these entries as described in [Adding a Directory Service Connection](#).

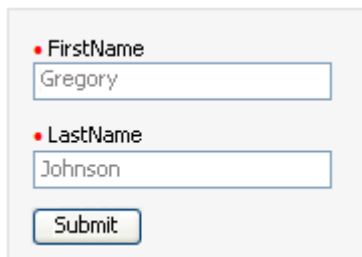
Deleting a Directory Service Connection

You can import a user/group into an AdminStudio Enterprise Server database using a Directory Service Connection. This will create a reference to this Directory Service Connection in the database. You can then import a Directory Service Attribute into the database.

In order to delete this Directory Service, user has to delete all the references to this Directory Service Connection. If there are references left (like users from this directory service or attributes from this directory service) you cannot delete this Directory Service Connection.

Managing Directory Services Attributes

When creating a Workflow Manager Template, you can assign a **Data Element** the **Data Type** of **Directory Service**. This means that when a Directory Service authenticated user performs the Workflow Step that requests that Data Element, information is pulled from the Directory Service to automatically populate the field, such as that user's name, email address, location, etc.



• FirstName
Gregory

• LastName
Johnson

Submit

Figure 4-1: Example of Fields Populated With Directory Services Attributes



Note • If the user is not connected using Directory Service authentication, then the fields will be left blank and enabled.

On the [Directory Services Attributes Administration Page](#), which is accessed by clicking **Manage Directory Services Attributes** on the **Directory Services** page, Administrators can map their customizations for the directory service.

Creating Directory Services Attributes

To add a new Directory Services attribute from the directory server into the AdminStudio Enterprise Server database, perform the following steps.



Task: *To create a new Directory Services attribute:*

1. On the **Directory Services** page, click **Manage Directory Services Attributes**. The **Directory Services Attributes Administration** page opens.
2. Click **Add**. The **Add Directory Service Attributes** view opens.
3. Using the current user's credentials, the **Attribute Name** list lists all of the Directory Services attributes from the Directory Services server. Select the Directory Services attribute that you want to add to the AdminStudio Enterprise Server database. A Directory Services attribute is used in Directory Services protocol to access information directories, such as `employeeNumber` or `documentAuthor`.
4. In the **Attribute Alias** text box, enter the name that you want to use to identify the Directory Services attribute in AdminStudio Enterprise Server, such as `Employee ID` or `Author`.
5. Click **Add** to add the defined Directory Services attribute. If AdminStudio Enterprise Server is unable to connect to the server (and no attributes are retrieved), the **Add** button is disabled.

Deleting Directory Services Attributes

To delete a Directory Services Attribute, open the **Directory Services Attributes Administration** page and click **Delete** next to the attribute that you want to delete.

Managing User Logins

This section explains the four methods of logging into AdminStudio Enterprise Server, and how to set the session timeout value.

- [Login Methods](#)
- [Using AdminStudio Enterprise Server Account Login](#)
- [Using Domain Account Login](#)
- [Using Single Sign-On Login](#)
- [Using Guest Account Login](#)
- [Setting the Session Timeout Value](#)

Login Methods

There are four methods you can use to login to AdminStudio Enterprise Server:

Table 4-1 • AdminStudio Enterprise Server Login Methods





Method	Description	Related Topics
AdminStudio Enterprise Server Account Login	Users login to AdminStudio Enterprise Server using a User Account (User Name and Password) that was created in AdminStudio Enterprise Server by clicking Add on the User Administration page.	<ul style="list-style-type: none"> Using AdminStudio Enterprise Server Account Login Creating a New User Account
Domain Account Login	<p>Users login to AdminStudio Enterprise Server using their Domain Account (User Name and Password), the same account they use to login to their network.</p> <p>To set this up, you import users or groups from a Directory Service: Active Directory or Novell eDirectory.</p>  <p>Note • For domain account login, the Anonymous access option on the IIS Authentication Methods Dialog Box must be selected.</p>	<ul style="list-style-type: none"> Using Domain Account Login Adding a Directory Service Connection Importing Directory Services Users and Groups
Single Sign-On Login	<p>You set up AdminStudio Enterprise Server so that if users are logged on to your network using their Domain Account (User Name and Password), they are automatically logged in to AdminStudio Enterprise Server.</p> <p>To set this up, you import users or groups from Active Directory</p>  <p>Note • For Single Sign-On, IIS performs user authentication. For all other login methods, AdminStudio Enterprise Server performs authentication.</p>  <p>Note • Single Sign-on is not supported for Novell eDirectory users and groups.</p>  <p>Note • For single sign-on login, the Anonymous access option on the IIS Authentication Methods Dialog Box must not be selected.</p>	<ul style="list-style-type: none"> Using Single Sign-On Login Adding a Directory Service Connection Importing Directory Services Users and Groups

Table 4-1 • AdminStudio Enterprise Server Login Methods (cont.)

Method	Description	Related Topics
Guest Account Login	You set up a Guest account to permit users without AdminStudio Enterprise Server login credentials to access features – such as viewing Reports or searching for an Application Request – without entering a User ID or Password.	<ul style="list-style-type: none">• Using Guest Account Login• Setting Up a Guest Account• Logging in as a Guest

IIS Authentication Methods Dialog Box

To set the Anonymous access option in IIS, perform the following steps.



Task: *To open the IIS Authentication Methods dialog box:*

1. Open Internet Information Services (IIS).
2. Select the AdminStudio Enterprise Server Web site in the tree view.
3. On the **Action** menu, select **Properties**. The **Web Site Properties** dialog box opens.
4. Open the **Directory Security** tab and click **Edit** in the **Anonymous access and authentication control** area. The **Authentication Methods** dialog box opens.

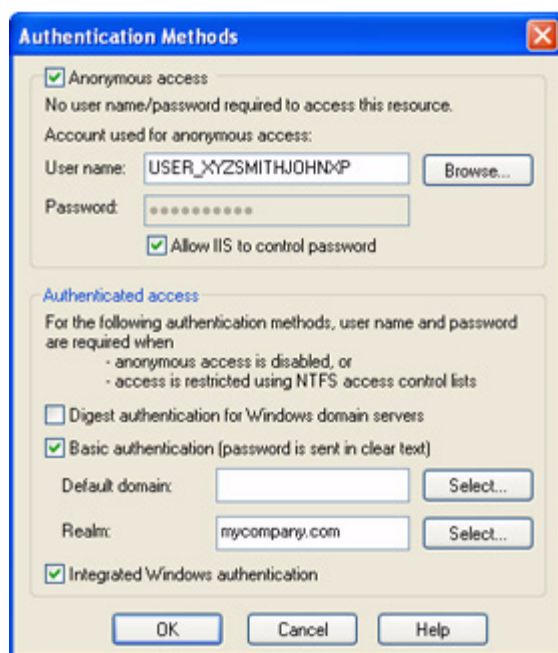


Figure 4-2: Anonymous Access option on the IIS Authentication Methods Dialog Box

Using AdminStudio Enterprise Server Account Login

To Set Up

To set up AdminStudio Enterprise Server User Account login, create a user account (**User Name** and **Password**) for each person that you want to have access to AdminStudio Enterprise Server. See [Creating a New User Account](#).

To Login

Users with AdminStudio Enterprise Server accounts enter their AdminStudio Enterprise Server-generated **User Name** and **Password** on the AdminStudio Enterprise Server Login page. AdminStudio Enterprise Server performs the authentication.

More Information

For more information, see [Creating a New User Account](#)

Using Domain Account Login

To Set Up

To set up Domain Account login, you do the following:

- Set up a **Windows Active Directory** or **Novell eDirectory** Directory Service Connection. See [Adding a Directory Service Connection](#).
- On your AdminStudio Enterprise Server IIS Server, open the [IIS Authentication Methods Dialog Box](#) and select the **Anonymous access** option. This specifies that AdminStudio Enterprise Server is going to authenticate the users.
- Import users or groups from that Directory Service. See [Importing Directory Services Users and Groups](#).

To Login

Imported users or users belonging to the imported groups enter their Directory Service **User Name** and **Password** on the AdminStudio Enterprise Server Login page.

AdminStudio Enterprise Server then connects with the Directory Service using the credentials provided on the [Add Directory Service Connection View](#), and then passes the login User Name and Password to the Directory Service so that it can authenticate the user.



Note • When entering the User Name, it is not necessary to enter the Directory Service domain name.

More Information

For more information, see:

- [Adding a Directory Service Connection](#)
- [Importing Directory Services Users and Groups](#)
- [IIS Authentication Methods Dialog Box](#)

Using Single Sign-On Login

To Set Up

To set up Single Sign-on login, you do the following:

- Set up a **Windows Active Directory** Directory Service Connection. See [Adding a Directory Service Connection](#).
- On your AdminStudio Enterprise Server IIS Web Server, open the [IIS Authentication Methods Dialog Box](#) and clear the **Anonymous access** option. This specifies that IIS is going to authenticate the users.
- Import users or groups from that **Active Directory** Directory Service. See [Importing Directory Services Users and Groups](#).



Note • Regarding Single Sign-On, note the following:

- For Single Sign-On, IIS performs user authentication. For all other login methods, AdminStudio Enterprise Server performs authentication.
- Single Sign-on is not supported for Novell eDirectory users and groups.

To Login

When a user with single sign-on opens AdminStudio Enterprise Server, the IIS Web Server checks to see if the user's domain **User Name** and **Password** are valid in the Active Directory Domain Server, and then automatically logs the user in to AdminStudio Enterprise Server. AdminStudio Enterprise Server does not need to connect to the Active Directory Domain Server.

For More Information

For more information, see:

- [Adding a Directory Service Connection](#)
- [Importing Directory Services Users and Groups](#)
- [IIS Authentication Methods Dialog Box](#)

Using Guest Account Login

This section describes how to setup a Guest Account and how to login using a Guest Account:

- [Setting Up a Guest Account](#)
- [Logging in as a Guest](#)

Setting Up a Guest Account

When a Guest Account is set up, AdminStudio Enterprise Server users without a User Account are automatically logged in using a Guest Account.

The AdminStudio Enterprise Server administrator can set up a Guest account to permit users without login credentials to access features such as viewing a report or searching for an Application Request . By using a Guest Account, Administrators do not have to add a new user account just to provide access for users who only need very limited functionality.

AdminStudio Enterprise Server system administrators should configure a Guest Account as follows:



Task:**To set up a Guest Account:**

1. Create a new user account to use as the Guest Account, and add this user to the `web.config` file located in the AdminStudio Enterprise Server Web Application `wwwroot` directory. Enter the user name in the following location of the `web.config` file:

```
<!-- Guest System Access -->
<add key="GuestAccount" value="username@companyname.com" />
```
2. On the **Administration** tab, click **Users & Groups**. The **User Administration** page opens.
3. Assign the Guest Account user to Roles with limited permissions. See [Managing Roles and Permissions](#) for more information.

Guest System Access Recommendations

When setting up Guest System Access, please follow these guidelines:

- The AdminStudio Enterprise Server Administrator should assign the Guest account user to Roles with limited privileges. If the Guest account belongs to a Role with high privileges, then every user who logs on as a Guest will have high privileges.
- If the `GuestAccount` key does not exist in the `web.config` file, then the **Guest Access** option does not appear on the AdminStudio Enterprise Server Login page.
- If the `GuestAccount` key exists in the `web.config` file but the user name specified in the value is empty (such as `value=""`), then the **Guest Access** option does not appear on the AdminStudio Enterprise Server Login page.
- If the `GuestAccount` key exists in the `web.config` file, but the user name specified in the value does not exist in the AdminStudio Enterprise Server database, then AdminStudio Enterprise Server displays an error message when a user tries to log on as a Guest User.

Configuring a Guest Account on a Workflow Manager Portal Site

Please note the following regarding configuring a Guest account on a Workflow Manager Portal site:

- If users are accessing Workflow Manager through a Workflow Manager portal, then the AdminStudio Enterprise Server administrator should add the GuestAccount key to the web.config file of the Portal site. By default, Workflow Manager will use the Guest account value configured at the portal site to log on the Guest user. If the portal site does not have any GuestAccount value configured, then Workflow Manager uses the GuestAccount value at the Workflow Manager Web Application site to log on the Guest account.
- If users are accessing the Workflow Manager site directly (not going through a portal), then Workflow Manager uses the GuestAccount value configured in the web.config file located in the AdminStudio Enterprise Server Web Application wwwroot directory to log on the Guest account.

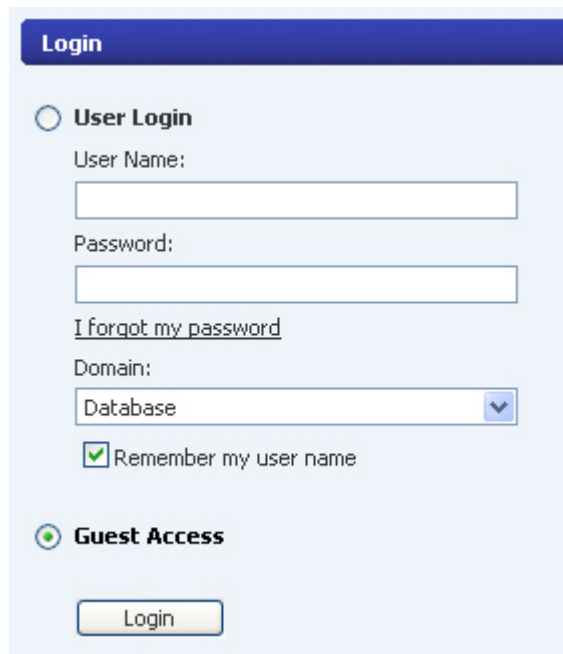


Note • The web.config file installed by AdminStudio Enterprise Server should contain the GuestAccount key with an empty value. It is the responsibility of the System Administrator to fill in that value in the web.config file of the Portal site or the AdminStudio Enterprise Server Web Application site.

Logging in as a Guest

Before anyone can log on to AdminStudio Enterprise Server as a Guest User, the System Administrator must configure a Guest Account, following the instructions in [Setting Up a Guest Account](#). Then a user could log on to your AdminStudio Enterprise Server site as a guest using the appropriate method:

- **If Directory Service authentication is used at your site**, and the user's Directory Service user name does not exist in the AdminStudio Enterprise Server database, AdminStudio Enterprise Server automatically logs this user on using the Guest account.
- **If Directory Service authentication is not used at your site**, AdminStudio Enterprise Server opens the AdminStudio Enterprise Server Login page by default. The user then selects the **Guest Access** option and clicks **Login**.



The screenshot shows a 'Login' dialog box with a blue header. It contains two main sections: 'User Login' and 'Guest Access'. The 'User Login' section is selected with a radio button. It includes fields for 'User Name', 'Password', and 'Domain' (a dropdown menu currently showing 'Database'). There is a link for 'I forgot my password' and a checked checkbox for 'Remember my user name'. The 'Guest Access' section is unselected. A 'Login' button is at the bottom.

The AdminStudio Enterprise Server Administrator assigns the Guest account to specific Roles, and the degree of permissions that the Guest account has depends upon which Roles the account is assigned to.

- **When a Role is created**, the administrator must indicate whether that Role has permission to create a Workflow Manager Application Request.
- **When Reports are created**, the administrator must indicate which Roles have permission to view them. Guest Users can view all Reports that their assigned Roles have permission to access.

Setting the Session Timeout Value

When AdminStudio Enterprise Server is installed, the session timeout value is set to 20 minutes. This means that after you login to the AdminStudio Enterprise Server, if your session remains idle for 20 minutes, you are prompted to login again.

You can change the session timeout value by editing two values in the web.config file: sessionState timeout and LoginPageTimeout.



Task: *To change the session timeout value:*

1. Open the wwwroot directory of the AdminStudio Enterprise Server installation and open web.config in a text editor.
2. Locate the line that contains sessionState timeout:

```
<sessionState timeout="20" mode="InProc" stateConnectionString= "tcpip=127.0.0.1:42424"
sqlConnectionString="data source=127.0.0.1; Trusted_Connection=yes" cookieless="false" />
```

3. Change the value specified for `sessionState timeout="nn"` to the new timeout value.
4. Locate the following line that contains `LoginPageTimeOut`:

```
<add key="LoginPageTimeOut" value="19.5" />
```
5. Change this value to match the timeout value you specified for the `sessionState`.

Users/Groups and Directory Services Reference

This section details the contents of the AdminStudio Enterprise Server pages that are used to manage users and directory services:

Table 4-2 • Users/Groups and Directory Services Tabs Reference

Page	Subpages
User Administration Page	<ul style="list-style-type: none"> • User Details View • Directory Services User/Group Add View
Directory Services Page	<ul style="list-style-type: none"> • Directory Services List Page • Add Directory Service Connection View • Directory Services Attributes Administration Page • Add Directory Service Attributes View

User Administration Page

This view lists all users defined in the system. From this view, you can add a new User, edit an existing User, or import a Directory Service User or Group.

Table 4-3 • User Administration Page Options

Option	Description
Add	Click to access the User Details View , where you can add a new User.
Directory Service User/Group Import	Click to access the Directory Services User/Group Add View , where you can import a User or Group from an Directory Service.
Status	To filter the list of users by Status, make a selection from this list: Active , Inactive or All .

Table 4-3 • User Administration Page Options (cont.)

Option	Description
User Name	Name user uses to log onto AdminStudio Enterprise Server, in the format of: <i>username@companyname.com</i> If a Directory Services Group has been imported, the Group name is listed in this column.
Roles	Lists the roles that this User has been assigned to.
Company	Name of user's Company.

From the **User Administration** page, you can perform the following tasks:

- **Create a AdminStudio Enterprise Server user account**—Create a user account for each person that you want to have access to AdminStudio Enterprise Server. See [Creating a New User Account](#).
- **Import Directory Service Users or Groups**—Set up a Windows Active Directory or Novell eDirectory Directory Service Connection and then click **Directory Service User/Group Import** on the **User Administration** page to import Users or Groups from that Directory Service. See [Importing Directory Services Users and Groups](#). Importing a group allows you to provide dynamic access to all of the members of that Group as the members of that Group changes.
- **Edit a user account**—To edit an existing account, click **Select** next to the **User Name** to open the **User Details** view.





Note • For more information on the methods for logging into AdminStudio Enterprise Server and how authentication is performed, see [Managing User Logins](#).

User Details View

On this view, enter or edit user account and imported Directory Services account information. On this page, you can also delete a user account.



Table 4-4 • User Details View Options

Option	Description
Company	Select the Company that this user or group belongs to.
User Name	<p>Perform one of the following:</p> <ul style="list-style-type: none"> • New—When creating a new user, enter the name the user will use to log onto AdminStudio Enterprise Server in the format of: <i>username@companyname.com</i>. • Importing—When importing a user group from a Directory Service, the name listed here will be prepopulated and uneditable.
Password	<p>Enter the User's password.</p>  <hr/> <p>Note • This field is not shown when importing a Directory Service Group.</p>
Status	Select Active to make this user or group active in the AdminStudio Enterprise Server system, or select Inactive to disable this account. See Disabling a User or Group Account .
Email	<p>User's email address.</p>  <hr/> <p>Note • This field only appears when creating a new user account. For users imported from a Directory Service, this information is retrieved dynamically from the Directory Service when needed.</p>
Roles	Listing of all of the defined Roles at the selected Company. Select the Roles appropriate for this user or group.
Location	Optionally, enter a geographic Location that you can use to group users together, such as New York Office or Midwest Region , etc. Assigning a geographical location to a user enables you to display this information in reports.
Update	Click to save your entries and close the User Details view.
Update and Add New	Click to save your entries and remain in the User Details view, where you can enter another new user.
Delete	Click to delete this user account.
Cancel	Click to exit this view without saving your entries.

Directory Services User/Group Add View

This view is accessed by clicking **Directory Service User/Group Import** on the **User Administration** page. The following options are included:

Table 4-5 • Directory Services User/Group Add View

Option	Description
Select a Directory Service	<p>Select the defined Directory Service that you want to import a user or group from.</p>  <p>Note • For information on defining a Directory Service Connection, see Adding a Directory Service Connection.</p>
Select Group or User	Specify whether you are importing a User or a Group .
Filter Directory Service List	<p>Select Show All to include all Users or Groups in the list, or select Filter list by, enter text in the box, and click Find to filter the list by the specified criteria.</p> <p>For example, to search for all of the users that start with the letter P, use the asterisk wildcard character (*) and enter P* in the Filter list by box.</p>
Select Group or User to Add	<p>From the Select Group or User to Add list, select the group or user to import.</p>  <p>Note • If you import a User that is a member of a Group that was previously imported, that User inherits the Roles that are assigned to that Group. You can then assign additional Roles to that User.</p>
Update	Click to save your entries.

The selected user or group is then opened in the [User Details View](#).

Directory Services Page

AdminStudio Enterprise Server can be integrated with Windows Active Directory and Novell eDirectory. This integration enables you to integrate Directory Services Users and Groups and also to use Directory Services attributes.

Table 4-6 • Directory Services Page Options

Option	Description
Manage Directory Service Connections	Click to open the Directory Services List Page where you can set up connections to Active Directory and eDirectory Directory Services.
Manage Directory Service Attributes	<p>Click to open the Directory Services Attributes Administration Page where you can map Directory Service attributes, which can be used when defining Data Elements.</p> <p>When a Data Element is defined as a Directory Service attribute, when this Data Element appears during a data entry step in a Workflow, information about the logged in user will be pulled from the Directory Service to populate those fields, such as: Department, Location, Employee Number, etc.</p>



Note • For more information about Directory Services and Lightweight Directory Access Protocol (LDAP), see [Ldp Overview](#) on the Microsoft TechNet Web site.

Directory Services List Page

Directory Services Connections are used to import users and groups into AdminStudio Enterprise Server, and to authenticate Active Directory or Novell eDirectory users logging into AdminStudio Enterprise Server.

You can choose to have a Directory Service listed in the **Domain** list on the AdminStudio Enterprise Server login page. This enables users in this Directory Service to login to AdminStudio Enterprise Server using their enterprise network credentials.

The information and options are included:

Table 4-7 • Directory Services List Page Options

Option	Description
Directory Service Name	The name that identifies this Directory Service in the Domain list on the AdminStudio Enterprise Server login page.
Description	A description of this Directory Service.

Table 4-7 • Directory Services List Page Options (cont.)

Option	Description
Directory Service Type	Identifies this Directory Service as either a Windows Active Directory or Novell eDirectory type.
Directory Service Host	The name or IP address of the Server on which this Directory Service is running.
Add	Click to add a new Directory Service connection.

Add Directory Service Connection View

On the **Add Directory Service Connection** view, which is opened by clicking **Add** on the [Directory Services List Page](#), you enter the necessary settings to connect to an Active Directory or Novell eDirectory Directory Service.

Table 4-8 • Add Directory Service Connection View Options



Option	Description
Use to Authenticate Users?	<p>Select Yes to include this Directory Service in the Domain list on the AdminStudio Enterprise Server login page. This enables users to login to AdminStudio Enterprise Server using their enterprise network credentials.</p>  <p>Note • For Single Sign-On, IIS performs user authentication. For all other login methods, AdminStudio Enterprise Server performs authentication.</p>
Directory Service Name	Enter a name to identify this Directory Service in the Domain list on the AdminStudio Enterprise Server login page. It is recommended that the Directory Service Name match the Domain Name.
Description	Enter a description of this Directory Service.
Directory Service Type	<p>Identify the type of Directory Service that you are adding a connection to:</p> <ul style="list-style-type: none"> • Active Directory • Novell eDirectory
Directory Service Host	Enter the name or IP address of the Server on which this Directory Service is running.
Directory Service Port	<p>Enter the port number of the Server on which this Directory Service is running. By default, AdminStudio Enterprise Server uses Port 389.</p>  <p>Important • The value for the Directory Service Port should be the port on the given server to which AdminStudio Enterprise Server should connect to send LDAP queries.</p>

Table 4-8 • Add Directory Service Connection View Options (cont.)


Option	Description
Base Distinguished Name	<p>Enter the Base Distinguished Name (DN) to identify the root node of this Directory Service.</p> <p>For example, for MyCompany, the Base DN could be:</p> <p>dc="MyCompany", dc="com"</p>
Domain Name	<p>Enter the Domain Name of this Directory Service. This Domain Name is used to create the right credentials, along with the Admin User Name. For example, you would enter DomainName\AdminUserName to connect to the specified Directory Service.</p>  <p>Note • To upgrade from AMS 2.5, enter the same Domain Name that was used in AMS 2.5. AdminStudio Enterprise Server will read this Domain Name and then try to authenticate the User using this Directory Service. For single sign-on to work on an upgrade scenario from 2.5 to 4.0, administrators have to configure at least one Directory Service.</p>
Use Secure Socket Layer (SSL)?	Select Yes if this Directory Service is configured to use Secure Socket Layer (SSL).
Connect Anonymously?	<p>Select No if you do not want to permit anonymous connections.</p> <p>If you select Yes to permit anonymous connections, AdminStudio Enterprise Server may not be able to authenticate Directory Service users and may not be able to add Directory Service users/groups into AdminStudio Enterprise Server.</p>
Admin Distinguished Name	<p>Enter the Distinguished Name of a user who has permission to retrieve users/groups information and authenticate any user against this Directory Service. Enter the Distinguished Name in the format of:</p> <p>domainname\username</p> <p>such as:</p> <p>abcinc\jsmith</p>
Password	Enter the password for the Admin Distinguished Name .
Group Class Name	<p>Enter the object class names for groups for this Directory Service. Default values are:</p> <ul style="list-style-type: none"> • Active Directory: group • eDirectory: groupofnames
Group Name Attribute	Enter an attribute used by this Directory Service to name groups. The default value for both Active Directory and eDirectory is cn.

Table 4-8 • Add Directory Service Connection View Options (cont.)

Option	Description
Group Member Attribute	Enter an attribute used by this Directory Service to define member groups. Default values are: <ul style="list-style-type: none"> • Active Directory: member • eDirectory: uniquemember
User Class Name	Enter the object class name used by this Directory Service for users. Default values are: <ul style="list-style-type: none"> • Active Directory: user • eDirectory: inetorgperson
User Name Attribute	Enter the attribute used by this Directory Service to identify users. Default values are: <ul style="list-style-type: none"> • Active Directory: samaccountname • eDirectory: uid
Update	Click to save your entries and return to the Directory Services List Page .
Update and Import (User/Group)	Click to save your entries and open the Directory Services User/Group Add View .
Test Connection	Click to test to see if the settings that you entered can be used to successfully connect to this Directory Service.

Directory Services Attributes Administration Page

On the **Directory Services Attributes Administration** page, which is accessed by clicking **Manage Directory Services Attributes** on the **Directory Services** page, Administrators can map their customizations for the directory services.

Directory Service attributes can be used when defining Data Elements. When a Data Element is defined as a Directory Service attribute, when this Data Element appears during a data entry step in a Workflow, information about the logged in user will be pulled from the Directory Service to populate those fields, such as: Department, Location, Employee Number, etc.

The following information is listed:

Table 4-9 • Directory Services Attributes Administration Page Options

Option	Description
Directory Service Attribute	Name of the LDAP (Lightweight Directory Access Protocol) Directory Service attribute that is used in Directory Services protocol to access information directories, such as employeeNumber or documentAuthor.

Table 4-9 • Directory Services Attributes Administration Page Options (cont.)

Option	Description
Attribute Alias	Name that you want to use to identify the Directory Service attribute in AdminStudio Enterprise Server, such as Employee ID or Author.
Delete	Click to delete the listed attribute.

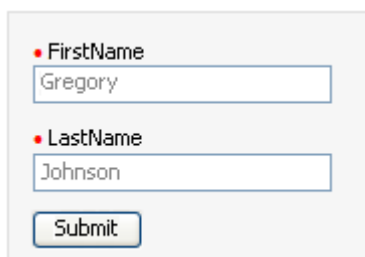
Add Directory Service Attributes View

When you click **Add** on the **Directory Services Attributes Administration** page to add a new Directory Service attribute into the AdminStudio Enterprise Server database, the following fields are displayed:

Table 4-10 • Add Directory Service Attributes View Options

Option	Description
Directory Service	Select the Directory Service that contains the attribute that you want to add.
Attribute Name	Lists all the attributes from the selected Directory Service server, using the currently logged user's credentials.
Attribute Alias	Enter the name that you want to use to identify the Directory Service attribute in AdminStudio Enterprise Server.
Add	Click to add the defined Directory Service attribute. If AdminStudio Enterprise Server is unable to connect to the server (and no attributes are retrieved), the Add button is disabled.

When a Data Element is defined as a Directory Service attribute during Workflow Manager Template creation, when this Data Element appears in the Workflow, information will be pulled from the Directory Service to populate those fields, such as:



The screenshot shows a web form with two text input fields. The first field is labeled 'FirstName' with a red dot bullet and contains the text 'Gregory'. The second field is labeled 'LastName' with a red dot bullet and contains the text 'Johnson'. Below these fields is a 'Submit' button.

However, if the user is not connected using Directory Service authentication, then the fields will be left blank and will be enabled.

Managing Roles and Permissions

All of the AdminStudio and Workflow Manager permissions are related to Roles. The amount of functionality that a user has is directly related to what Roles they are assigned to.

This section contains information on the following tasks:

- [AdminStudio and Workflow Manager Roles and Permissions](#)
- [Creating a New Role](#)
- [Editing an Existing User Role](#)
- [Copying an Existing Role](#)
- [Roles Reference](#)

AdminStudio and Workflow Manager Roles and Permissions

Permissions are assigned to Roles, and Roles are assigned to users. Therefore, all permissions in AdminStudio and Workflow Manager are controlled by Roles.

This section includes the following topics:

- [Role Permission Lists](#)
 - [AdminStudio Client Tools Permissions](#)
 - [AdminStudio Enterprise Server Tools Permissions](#)
 - [Workflow Manager Permissions](#)
- [System Roles](#)

Role Permission Lists

Permissions to perform all AdminStudio and Workflow Manager functions are assigned using Roles. A user has permission to perform only those tasks that are explicitly selected in the Role(s) that the user is assigned to.

This section describes all of the AdminStudio and Workflow Manager permissions:

- [AdminStudio Client Tools Permissions](#)
- [AdminStudio Enterprise Server Tools Permissions](#)
- [Workflow Manager Permissions](#)

AdminStudio Client Tools Permissions

The AdminStudio client tools have the following permissions:

Table 5-1 • AdminStudio Client Tools Permissions

Category	Permission
AdminStudio Projects Tab	<ul style="list-style-type: none"> • Edit—Permits user to view and edit those Projects on the Projects tab that are assigned to him. • Create—Permits user to create new Projects and assign them to users. Users with the Create permission see a list of all users and their associated Projects on the Projects tab. Users with only the Edit permission cannot create new Projects and can only view and edit Projects that are assigned to him. • Delete—Permits user to delete a Project from the Projects tab.
AdminStudio Workflow Templates Tab	<ul style="list-style-type: none"> • View—Permits user to view the Workflow tab and all of the existing Workflows. • Edit—Permits user to edit existing Workflows on the Workflow tab. • Create—Permits user to create a new Workflow on the Workflow tab. • Delete—Permits user to delete a Workflow from the Workflow tab.
AdminStudio Tools Tab	<ul style="list-style-type: none"> • Add—Permits user to add a new tool to the Tools tab. • Edit—Permits user to edit the properties of an existing tool on the Tools tab. • Delete—Permits user to delete an existing tool on the Tools tab.

Table 5-1 • AdminStudio Client Tools Permissions (cont.)

Category	Permission
AdminStudio General	<ul style="list-style-type: none"> • Modify Tools Options Dialog—Permits user to set options on the Locations, Updates and Quality tabs of the AdminStudio Options dialog box. Users without this permission can view the Options dialog box but cannot make any changes. If a user has only the Modify Tools Options Dialog permission but does not also have the Change Default Application Catalog permission, that user cannot edit the Shared AdminStudio Application Catalog field on the Application Catalog tab of the Options dialog box. • Change Default Application Catalog—Permits user to edit the Shared AdminStudio Application Catalog field on the Application Catalog tab of the Options dialog box (if that user also has the Modify Tools Options dialog permission). This permission also permits user to edit the Make this the default shared Application Catalog option on the Connect Application Catalog dialog box
Application Manager / Conflict Solver Conflicts	<ul style="list-style-type: none"> • Run Analysis—Permits user to perform conflict analysis on a package. • Resolve—Permits user to resolve any automatically resolvable conflicts found during conflict analysis on a package. • Modify Rules—Permits user to open the Rules Viewer and create and edit new Rules. • Modify Data—Permits user to create new Groups, rename existing Groups, and modify Group Properties. Permits user to Cut and Paste a Group to a new location. Permits user to Copy/Cut and Paste a package into a new Group. Permits user to modify options on the Resolution Options dialog box. Permits user to edit a package Description on the Products View.
Application Manager / Conflict Solver Package	<ul style="list-style-type: none"> • Delete—Permits user to delete a package from the Application Catalog. • Import—Permits user to import a package into the Application Catalog. • Modify Extended Attributes—Permits user to modify a package's metadata on the Extended Attribute view. • Delete History—Permits user to delete a package's history log (which contains a record of any operation that materially changes a software package or the data associated with it).

Table 5-1 • AdminStudio Client Tools Permissions (cont.)

Category	Permission
Application Manager / Conflict Solver Other	<ul style="list-style-type: none"> • Modify Tools Options Dialog—Permits user to open the Application Manager/ConflictSolver Options dialog box and set options. • Change Application Catalog—Permits the user to connect to a different Application Catalog by selecting Connect on the Catalog menu. Without this permission, this selection is disabled. • Run Merge Wizard—Permits user to use the Merge Wizard to merge the application data of a specified Application Catalog into the currently open Application Catalog. • Run Validation—Permits user to use the Validation Wizard to validate a Windows Installer package against custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards. • Scan for Dependencies—Permits user to generates a list of all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog. • Run Directory Monitoring—Permits user to use the Network Directory option of Package Auto Import to monitor a directory location on the network (or a local directory) and automatically import or re-import packages in that directory at scheduled intervals. • Run Best Practices—Permits user to use the Conflict Wizard or the ConflictSolver Process Assistant to evaluate source packages to see if they meet Windows Installer best practices rules.
Patch Impact Manager	<ul style="list-style-type: none"> • Import Patch—Permits user to use OS Security Patch Wizard to import Windows operating system patches into the Application Catalog. • Run Analysis—Permits user to use the ConflictSolver Patch Impact Analysis Wizard to analyze the impact of installing an OS Security patch on user machines.
Software Repository	<ul style="list-style-type: none"> • Overwrite—Permits user to import a duplicate package into the Software Repository, overwriting the existing version.

AdminStudio Enterprise Server Tools Permissions

The AdminStudio Enterprise Server tools have the following permissions:

Table 5-2 • AdminStudio Enterprise Server Tools Permissions

Category	Permission
People	<ul style="list-style-type: none"> • View—Permits user to see the Administration/Users & Groups tab. • Add—Permits user to add new users and import a users list from Active Directory. • Edit—Permits user to modify existing users. • Delete—Permits user to delete existing or newly created users.
Roles	<ul style="list-style-type: none"> • View—Permits user to see the Administration/Roles tab. • Add—Permits user to add new Roles. • Copy—Permits user to copy an existing Role. • Edit—Permits user to modify an existing Role. • Email Notify Enabled—<i>Not currently in use.</i>
Directory Services	<ul style="list-style-type: none"> • View—Permits user to see the Administration/Directory Services tab and view the Directory Services page, the Directory Services List page, and the Directory Services Attributes Administration page. • Add—Permits user to add a new Directory Service. • Edit—Permits user to edit an existing Directory Service. • Delete—Permits user to delete existing or newly created Directory Services.
System Settings	<ul style="list-style-type: none"> • View—Permits user to open the Administration/System Settings tab. • Edit—Permits user to edit the fields on the Administration/System Settings tab: AdminStudio Enterprise Server User Account and the Workflow Manager/ AdminStudio Enterprise Server Serial Numbers.

Table 5-2 • AdminStudio Enterprise Server Tools Permissions (cont.)

Category	Permission
Job Manager	<ul style="list-style-type: none"> View—Permits user to see the Job Manager tab only if that user also has at least one of the following Job Manager permissions: Manage Jobs, Manage Templates, or Configuration. If you only have the Job Manager View permission (and none of the other Job Manager permissions), you will not be able to view the Job Manager tab or any of its subtabs. Manage Templates—Permits user to see the Job Manager/Manage Templates tab, view the list of Templates on the Job Template List page, and view the details of an existing Template on the Edit Template page. Users with only the View and Manage Templates permission cannot create a new Template or edit an existing Template. When a Role is assigned the Manage Templates permission, the View permission is automatically selected. Manage Jobs—Permits user to see the Job Manager/Manage Jobs tab, view the list of Jobs on the Jobs Queue page, and view the details of a Job that they created on the Edit Job page. Users with only the View and Manage Jobs permission cannot create a new Job or edit an existing Job. When a Role is assigned this permission, the View permission is automatically selected. Manage Jobs for All Owners—Adds the Show All Jobs check box option to the Jobs Queue page, which enables this user to see a list of Jobs created by all users. For users without this permission, only those Jobs that they created are listed. Configuration—Permits the user to view the list of conflict detection rules that are selected on the Conflict Detection Options page. However, if this user does not also have the Edit permission, this user will not be able to make any changes to this list. When a Role is assigned this permission, the View permission is automatically selected. Edit—If a user also has the Manage Templates and/or Manage Jobs permissions, this permission allows the user to create new Templates and Jobs, and also to edit existing Templates and Jobs. If this user has this permission and also has the Configuration permission, this user can modify the selections on the Conflict Detection Options page.

Table 5-2 • AdminStudio Enterprise Server Tools Permissions (cont.)

Category	Permission
Report Center / All Reports	<ul style="list-style-type: none"> • Add—Permits user to click Add on the All Reports page and create a new Custom Report using the Report Wizard. • Edit—Permits user to edit an existing Custom Report. • Delete—Permits user to delete an existing Custom Report.
Report Center / Package Reports	<ul style="list-style-type: none"> • View—Permits user to see the Package Reports tab and view Package Reports.
Software Distribution	<ul style="list-style-type: none"> • Operate—Permits user to use Configuration Manager Web Console (the Distribute tab of AdminStudio Enterprise Server) to configure package distribution options and distribute setup packages without going to Configuration Manager Server.

Workflow Manager Permissions

Workflow Manager has the following permissions:

Table 5-3 • Workflow Manager Permissions

Category	Permission
Application Request	<ul style="list-style-type: none"> • View—Permits user to view the Properties and Issues tabs on the Application Progress page of existing Application Requests. • Add—Permits user to create a new Application Request. • Edit—Permits user to edit an existing Application Request. • Copy—Permits user to copy an existing Application Request. • Delete—Permits user to delete an existing Application Request. • Monitor Application Progress—Permits user to see the Application Progress, Uploaded Files, and Downloadable Files tabs of the Application Progress page. Also permits user to complete Workflow Steps. • View Related Applications—Permits user to view the Related Applications tab of the Application Progress page. • Add Related Applications—Permits user to link one Application Request to another on the Related Applications tab. • Delete Related Application Links—Permits user to delete Related Application links. • Edit Application Due Period—Permits user to edit the Application Due Period field on the Properties tab of the Application Progress page.

Table 5-3 • Workflow Manager Permissions (cont.)

Category	Permission
Application Status Management	<ul style="list-style-type: none"> • View—Permits user to view the Application Status Administration page and to view (but not update) Application Status details on the Edit Application Status view. • Add—Permits user to add a new Application Status. • Edit—Permits user to update Application Status details on the Edit Application Status view. • Delete—Permits user to delete an existing Application Status
Calendar Settings	<ul style="list-style-type: none"> • View—Permits user to view the Calendar Settings Administration page. • Edit—Permits user to edit the settings on the Calendar Settings Administration page.
Issues	<ul style="list-style-type: none"> • View—Permits user to view the Issues tab of the Application Progress page and to view the issue details. • Add—Permits user to create a new E-mail or Knowledge Base issue . • Respond—Permits user to respond to an existing issue. • Close—Permits user to close an E-mail or Knowledge Base issue. • Add/Close Critical Issue—Permits user to create and close Critical issues.
My Notifications	<ul style="list-style-type: none"> • View—Permits user to view the My Notifications pages: My Default Project Notifications and My Application Notifications. • Edit—Permits user to edit the settings on the My Notifications pages: My Default Project Notifications and My Application Notifications.
Project Permissions	<ul style="list-style-type: none"> • View—Permits user to view (but not edit) the permission and email settings on the Project Permission and Notification Settings view. • Edit—Permits user to edit the permission and email settings on the Project Permission and Notification Settings view.
Projects	<ul style="list-style-type: none"> • View—Permits user to view the Project Administration page and view (but not update) Project details on the Project Details view. • Add—Permits user to create a new Project. • Edit—Permits user to edit the details of an existing Project. • Delete—Permits user to delete a Project.
Requestor Company	<ul style="list-style-type: none"> • View—Permits user to view a list of existing Requestor companies and view Requestor company details. • Add—Permits user to add new Requestor companies. • Edit—Permits user to update Requestor company details. • Delete—Permits user to delete an existing Requestor company.

Table 5-3 • Workflow Manager Permissions (cont.)

Category	Permission
Search	<ul style="list-style-type: none"> • Simple Search—Permits user to enter keywords in the Application Search box on the Workflow Manager Home page to search for an Application Request by name. • Advanced Search—Permits user to search for Application Requests by multiple criteria on the Application Search page.
Servicer Company	<ul style="list-style-type: none"> • View—Permits user to view a list of existing Servicer companies, view Servicer company details, and update those details. • Add—Permits user to create new Servicer companies.
Templates	<ul style="list-style-type: none"> • View—Permits user to view the Template Administration page and view Template details on the Template Details page. Also gives you permission to add an external data source on the External Data Sources subtab of the Administration tab. • Add—Permits user to edit an existing Template and create a new Template. • Copy—Permits user to copy an existing Template.
Terminology	<ul style="list-style-type: none"> • View—Permits user to view and modify AdminStudio Enterprise Server system terminology.
Work Assignment	<ul style="list-style-type: none"> • View—Permits user to view the Working Queue View and Work Assignment Listing pages to see a list of existing work assignments. • Assign—Permits user to assign work on the Assignment Details page.
Global Email Administration	<ul style="list-style-type: none"> • View—Permits user to view the Email Settings Administration and External Email Address Administration pages. • Edit—Permits user to edit the settings on the Email Settings Administration and External Email Address Administration pages.
Deployment Sites	<ul style="list-style-type: none"> • View—Permits user to see the Deployment Site Administration page. • Add—Permits user to add a new Deployment Site. • Edit—Permits user to edit an existing Deployment Site.

System Roles

AdminStudio and Workflow Manager come with default **System Roles** which cannot be modified. These Roles were created based upon the needs of typical AdminStudio and Workflow Manager users, providing each Role with only the permissions a user would need to perform their required tasks. You can either assign these **System Roles** to users, or you can copy **System Roles** and then modify them to customize them for your organization.

Copied System Roles or new Roles that you create have a **Role Type** of **User** (User Roles), while default Roles created during installation have a **Role Type** of **System** (System Roles). A Role's **Role Type** is listed on the **Role Administration** page and cannot be changed.

Information about System Roles is organized in the following sections:

- [Superuser Role: AMSSuper](#)
- [Default System Roles](#)
- [Default System Users](#)

Superuser Role: AMSSuper

When AdminStudio Enterprise Server and Workflow Manager are installed, the following user is created and is assigned the **AMSSuper** Role:

- **User Name:** suams
- **Password:** suams

A user assigned the **AMSSuper** Role can create Servicer companies and customize terminology. All other tasks should be performed by a Service Company administrator or by accounts created by a Service Company administrator.



Note • The **AMSSuper** Role is not listed on the Role Administration page unless you are logged on using the **suams** account.

Default System Roles

When AdminStudio Enterprise Server and Workflow Manager are installed, the following System Roles are created:

Select	Role Name	Company Name	Role Description	Role Type
> Select	Application User	Requester	Application User for Requestor Company	System
Select	Project Manager	Requester	Project Manager for Requestor Company	System
Select	Project Manager	Servicer	Project Manager Role	System
Select	Repackager	Servicer	Application Repackager Role	System
Select	SCAdmin	Servicer	System Administrator Role	System
Select	Tech Lead	Servicer	Tech Lead Role	System
Select	UA Tester	Requester	User Acceptance tester for Requestor Company	System

Figure 5-1: Default System Roles



Note • If you did not purchase Workflow Manager, the Roles associated with the Requester company are not listed: **Application User**, **Project Manager**, and **QA Tester**.



Note • If you are connected to an Application Catalog that has been upgraded from AMS 2.5, all Roles will have a **Role Type** of **User**.

The following table lists the default System Roles:

Table 5-4 • Default System Roles

Role Name	Company Name	Description
Project Manager	Servicer	Project Manager for Servicer Company
Repackager	Servicer	Application Repackager for Servicer Company
SCAdmin	Servicer	System Administrator for Servicer Company
Tech Lead	Servicer	Tech Lead for Servicer Company
Application User	Requester	(Workflow Manager only) Application User for Requester Company
Project Manager	Requester	(Workflow Manager only) Project Manager for Requester Company
UA Tester	Requester	(Workflow Manager only) User Acceptance Tester for Requester Company

You can view the permissions of each of these Roles by selecting the Role on the **Role Administration** page, and then expanding the **Role Permissions** list.

Default System Users

When AdminStudio Enterprise Server and Workflow Manager are installed, a User Account is created for each of these System Roles:

Select	User Name	Roles	Company
Select	admin@servicer.com	SCAdmin	Servicer
Select	pm@requester.com	Project Manager	Requester
Select	pm@servicer.com	Project Manager	Servicer
Select	repackager@servicer.com	Repackager	Servicer
Select	techlead@servicer.com	Tech Lead	Servicer
Select	tester@requester.com	UA Tester	Requester
Select	user@requester.com	Application User	Requester

Figure 5-2: Default System Users

To see what functionality is available for one of these default System users, you could log on as that user and then observe what features are available. You could also select the System user's associated Role on the **Role Administration** page, and then expand the **Role Permissions** list.



Note • By default, the password for each of these default System Users is the same as the text prior to the @ sign (such as **techlead** for techlead@servicer.com), except the password for pm@requester.com and pm@servicer.com is **projectmanager**.

Creating a New Role

To create a new Role, perform the following steps.



Task: *To create a new Role:*

1. On the **Administration** tab, click **Roles**. The **Role Administration** page opens.
2. Click the **Add** button. The **Role Details View** opens.
3. Enter a name in the **Role Name** field to uniquely identify this Role.
4. From the **Role Company** list, select the name of the Company whose users can be assigned this Role.
5. In the **Role Description** text box, describe the purpose of this new Role.
6. In the **Role Permissions** list, select the permissions you want to assign to this Role. For a detailed list of the available permissions, see [Role Permission Lists](#).
7. Click the **Update** button. The new Role now appears in the list on the **Role Administration** page.

Editing an Existing User Role

You can edit all Roles with a **Role Type** of **User**, but System Roles cannot be modified.



Task: *To edit an existing User Role:*

1. On the **Administration** tab, click **Roles**. The **Role Administration** page opens.
2. Select the Role that you want to edit. The **Role Details View** opens.
3. Make edits to the **Role Name** and **Role Description** fields, if desired.



Note • *It is not recommended that you change a Role's **Role Company**.*

4. In the **Role Permissions List**, edit the permissions assigned to this Role.
5. Click the **Update** button. You are returned to the **Role Administration** page.

Copying an Existing Role

You can make a copy of an existing User or System Role using a new Role Name. You can then modify it to customize it for your organization. You cannot modify a System Role. You can also copy an existing Role from one Company to another.

To copy an existing Role, perform the following steps.



Task: *To copy an existing Role:*

1. On the **Administration** tab, click **Roles**. The **Role Administration** page opens.
2. Click the **Copy** button. The **Copy Role View** opens.
3. From the **Copy from Company** list, select the name of the Company whose Role you want to copy.
4. From the **Copy from Role** list, select the Role that you want to copy.
5. From the **Copy to Company** list, select the Company that you are creating this new Role for.
6. Enter a name in the **New Role Name** field to uniquely identify this Role.



Note • You are not permitted to have two Roles in the same Company with the same name. You are permitted to use the same Role name in more than one Company.

7. From the **Role Company** list, select the name of the Company whose users can be assigned this Role.
8. In the **Role Description** text box, enter a description for the Role.
9. In the **Role Permissions List**, edit the permissions that were previously assigned to this Role.
10. Click the **Update** button. The new Role now appears in the Role list on the **Role Administration** page.

Deleting a Role

You can delete all Roles with a **Role Type** of **User**, but **System** Roles cannot be deleted.



Task: *To delete an existing User Role:*

1. On the **Administration** tab, click **Roles**. The **Role Administration** page opens.
2. Select the user-defined Role that you want to delete. The **Role Details View** opens.
3. Click **Delete**. You are prompted to confirm the deletion.
4. Click **OK**. The **Role Details View** closes and the Role you deleted is no longer listed on the **Role Administration** page.

Roles Reference

Reference information for Roles is presented in the following sections:

- [Role Administration Page](#)
- [Copy Role View](#)
- [Role Details View](#)

Role Administration Page

The **Role Administration** page lists all defined Roles on the system. From this view, you can edit existing Roles or add new ones. The following options are available:

Table 5-5 • Role Administration Page

Option	Description
Select	Click to access the Role Details View , where you can edit this Role.
Role Name	Name of Role.
Role Company	Name of company that this Role was defined for.
Role Description	Description of the purpose of this Role.
Role Type	<p>Identifies this role as one of the following:</p> <ul style="list-style-type: none"> • System—Role was created during installation and cannot be deleted or modified. However, it can be copied and modified to create a User Role. • User—Role created by an administrator either by copying an existing System Role or by creating a new Role. <p>For more information, see System Roles.</p>
Add	Click to access the Role Details View , where you can add a new Role.
Copy	Click to access the Copy Role View , where you can copy an existing Role.

Copy Role View

On the **Copy Role** view of the **Role Administration** page you can copy an existing System or User Role.

Table 5-6 • Role Administration Page / Copy Role View Options


Option	Description
Copy from Company	Select the name of the Company that has a Role that you want to copy.
Copy from Role	Select the name of the Role that you want to copy. You can copy either System or User Roles.
Copy to Company	Select the name of the Company that this new Role is being created for.
New Role Name	Enter a name to identify this new Role.
Copy	Click to create this new Role and close the Copy Role view.
Cancel	Click to exit this view without creating a new Role.

Role Details View

Permissions are assigned to Roles, and Roles are assigned to users. Therefore, all permissions in AdminStudio and Workflow Manager are controlled by Roles.

On the Role Details view, you can enter or edit information to identify a Role and set the Role Permissions. You can also delete a user-defined Role.

Table 5-7 • Role Administration Page / Role Details View Options

Option	Description
Role Name	Enter a name to uniquely identify this Role
Role Company	Select the Company whose employees can be assigned the Role.
Role Description	Enter text to identify the purpose of this Role.
Role Permissions List	Select the specific functionality permissions that you want to assign to this Role. For more information, see Role Permission Lists and System Roles .
Update	Click to save your entries and close the Role Details view.
Delete	<p>Click to delete this user-defined Role.</p>  <p>Note • If you are editing a System Role, the Delete button will be disabled.</p>
Cancel	Click to exit this view without saving your entries.

Managing Application Catalog Databases



Edition • Application Manager is included with AdminStudio Professional and Enterprise Editions.

You use Application Manager to manage Application Catalog databases. Tasks you perform using the Application Manager include importing packages, merging application catalogs, setting up automatic package import, organizing packages into groups, and viewing a package's extended attributes, files/components, INI file changes, registry entries, shortcuts, merge modules, history, and tables.

Information about using Application Catalogs is organized into the following sections:

Table 6-1 • Topics Regarding Using Application Catalogs


Section	Description
Application Manager and ConflictSolver	Describes the functionality available using Application Manager and how it differs from ConflictSolver.
Managing Application Catalogs	Explains how to create, connect to, search for, organize, view information about, and delete packages from.
Importing Data	<p>Explains how to import Windows Installer packages (.msi) with any associated transforms (.mst) or patches (.msp), merge modules (.msm), OS snapshots (.osc), Marimba NCP files (.ncp), and other setup types.</p> <div>  <p>Edition • Import support for Marimba Native Channel Packager (.ncp) files is available in AdminStudio Enterprise Edition.</p> </div>
Using the Software Repository	Explains how to manage a software package's associated installation files using the Software Repository.

Table 6-1 • Topics Regarding Using Application Catalogs (cont.)

Section	Description
Sharing Application Catalog Data	Describes the three methods for sharing Application Catalog data in your enterprise: merge, auto-import, and replication.
Taking OS Snapshots	Explains how to create OS Snapshots using the OS Snapshot Wizard.
Reference	Describes every view, dialog box, and Wizard available in Application Manager.

Application Manager and ConflictSolver

Application Manager and ConflictSolver are closely related tools. Both provide a comprehensive view of the Application Catalog. However, the tasks you can perform using each of these tools differ:

- **ConflictSolver** is used to perform conflict detection and resolution.
- **Application Manager** is used for importing and storing packages in a database, and organizing those packages.

This section compares these tools and includes the following topics:

- [Functionality Comparison](#)
- [Accessing the Tools](#)
- [Displaying ConflictSolver Options in Application Manager](#)
- [Creating and Connecting to Application Catalogs in Application Manager and ConflictSolver](#)

Functionality Comparison

This table compares the functionality included in Application Manager and ConflictSolver.

Table 6-2 • Comparison of ConflictSolver and Application Manager Functionality


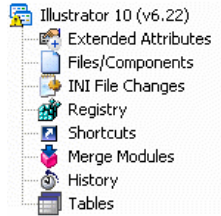

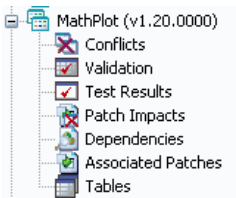
View	Major Functionality	Package Sub-Nodes
Application Manager 	<p>Use to perform the following tasks</p> <ul style="list-style-type: none"> • Import packages—Import packages into the Application Catalog. • Manage packages—Organize packages into Groups, rename packages, and delete packages. • Merge Wizard—Merge data from a source Application Catalog into the currently open Application Catalog. • Package Auto Import—Set up the automatic import of packages. • Software Repository—Import packages into and check in and check out packages from the Software Repository. • Associate a package with a Workflow Manager package • Launch applications—Launch Predeployment Test Preparation Wizard, InstallShield Editor, and Distribution Wizard. • Reports—View Package, File, and Registry reports. 	

Table 6-2 • Comparison of ConflictSolver and Application Manager Functionality

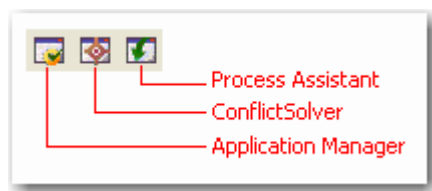
View	Major Functionality	Package Sub-Nodes
ConflictSolver 	<p>Use ConflictSolver to perform the following tasks:</p> <ul style="list-style-type: none"> • Conflict Analysis and Resolution—Check one or more packages for conflicts against others in the Application Catalog, and resolve those conflicts. • Validation—Validate Windows Installer package against Microsoft ICEs (Internal Consistency Evaluators), • Test Results—View PackageExpert test results. • Patch Impact Analysis Wizard—Launch Patch Impact Analysis Wizard for selected package. • Dependency Scan—Generate a list of all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog. • Associated Patches—View a list of imported patches that, if installed, would update the selected product. • Reports—View Package, File, and Registry reports. <p>For information on using ConflictSolver, see Identifying and Resolving Application Conflicts Using ConflictSolver.</p>	

Accessing the Tools

Both Application Manager and ConflictSolver can be opened from the Start Page, the Tools Gallery, or from a menu shortcut.



You can toggle between Application Manager, ConflictSolver, and ConflictSolver Process Assistant by clicking a button in the toolbar:

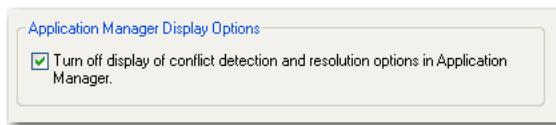


When the ConflictSolver or Application Manager is active, **ConflictSolver** or **Application Manager** is listed in the Title bar and also in the top right corner of the interface.

Displaying ConflictSolver Options in Application Manager

By default, all functionality for interacting with the Application Catalog, including conflict detection and resolution, is available in Application Manager.

To turn off the display of conflict detection and resolution options in Application Manager, select the check box in the **Application Manager Display Options** area of the **General** tab of the ConflictSolver and Application Manager **Options** dialog box.



Creating and Connecting to Application Catalogs in Application Manager and ConflictSolver

You can connect to Application Catalogs from both Application Manager and ConflictSolver as described in [Connecting to an Existing Application Catalog](#). However, you use the AdminStudio interface to create new application catalogs. See [Creating New Application Catalogs](#).

Managing Application Catalogs

Package information is stored in Application Catalogs. When you use Application Manager, you will either automatically be logged into an Application Catalog, or you will have to connect to an existing catalog. Once you have connected to a catalog, you can manipulate data in Application Manager, including creating and organizing groups and entering extended attribute data.

The following topics relate to Application Catalog management:

- [About AdminStudio Application Catalogs](#)
- [Using the AdminStudio Quick Start Evaluation Experience](#)
- [Connecting to an Existing Application Catalog](#)
- [Creating New Application Catalogs](#)
- [Specifying a Default AdminStudio Application Catalog](#)
- [Searching an Application Catalog](#)
- [Disconnecting from an Application Catalog](#)

- [Upgrading an Existing Application Catalog](#)
- [Organizing Your Application Catalog Using Groups](#)
- [Viewing and Editing Package Extended Attributes](#)
- [Viewing Package History](#)
- [Deleting Packages](#)

About AdminStudio Application Catalogs

This section provides the following information:

- [Overview of Application Catalogs](#)
- [Standalone Application Catalog vs. the AdminStudio Enterprise Server Application Catalog](#)

Overview of Application Catalogs

To perform enterprise level data checking, it is very efficient if the data can be consolidated into a single database. The AdminStudio Application Catalog database consolidates data from many installation packages in a single location.

Application Catalogs can contain the complete contents of Windows Installer (.msi) packages, Merge Module (.msm) packages, and OS snapshots (.osc). Depending upon the import options that are set, an Application Catalog can also include binary records. Application Catalogs can also store application, workflow, permissions, Predeployment Test, Microsoft Patch, PackageExpert, and Workflow Manager data.

You can use ConflictSolver and PackageExpert to test packages in the Application Catalog:

- **ConflictSolver**—You can use ConflictSolver's MSI ICEs (internal consistency evaluators) to evaluate specific packages, and can use ConflictSolver ACEs (application conflict evaluators) to extend the ICEs and evaluate data across packages.
- **PackageExpert**—You can use PackageExpert to analyze the integrity of packages in the Application Catalog and also identify and resolve any issues packages would have when deployed on the Windows Vista operating system.

Sharing Application Catalogs

In multi-user environments, the AdminStudio Application Catalog is typically shared. The AdminStudio Administrator can make a shared catalog the default catalog, which results in all AdminStudio users who use the same shared location using the same shared Application Catalog. If the shared Application Catalog is changed, users will automatically open up the new shared Application Catalog. For more information, see [Specifying a Default AdminStudio Application Catalog](#).

Sharing Application Catalog Data

There are three methods for sharing Application Catalog data in your enterprise:

- **Merge Wizard**—Use to merge data from a source Application Catalog into the currently open Application Catalog.
- **Package Auto Import Wizard**—Offers two methods of automatically importing Windows Installer packages into your Application Catalog and keeping them updated.
- **Application Catalog Replication**— Enables you to distribute an Application Catalog database where it is needed across a large enterprise or multi-national corporation.

See [Sharing Application Catalog Data](#) for detailed information.

Standalone Application Catalog vs. the AdminStudio Enterprise Server Application Catalog



Edition • AdminStudio Enterprise Server Tools are included with AdminStudio Enterprise Edition.

You can connect to any standalone SQL Server Application Catalog, or you can connect to the AdminStudio Enterprise Server Application Catalog:

- **Standalone**—A Standalone Application Catalog is not associated with the Enterprise Server tools. The AdminStudio client tools connect directly to the database server hosting the standalone Application Catalog.
- **AdminStudio Enterprise Server**—All of the AdminStudio Enterprise Server tools—Security Console, Report Center, Job Manager, and Workflow Manager—are connected to the AdminStudio Enterprise Server Application Catalog. AdminStudio client tools can also connect to this Application Catalog, allowing you to store all data generated on a package in the same location, and to link packages in this Application Catalog to Workflow Manager Application Requests.



Note • AdminStudio Enterprise Tools are always connected to the AdminStudio Enterprise Server database.

For more information on using Application Catalogs, see the following:

- [Connecting AdminStudio Client Tools to a Standalone Application Catalog](#)
- [Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog](#)

Using the AdminStudio Quick Start Evaluation Experience

When you initially open AdminStudio, because a default Application Catalog has not yet been set, the **Default Application Catalog** dialog box opens, prompting you to open an Application Catalog.

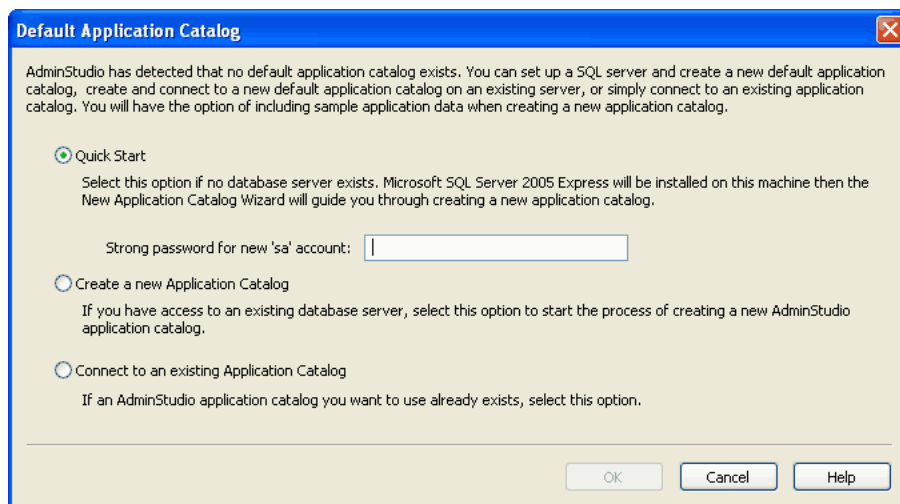


Figure 6-1: Default Application Catalog Dialog Box

To quickly get started using AdminStudio, select one of the following options:

- **If you do not have access to a database server**, but you want to quickly start using AdminStudio, select the **Quick Start** option. Microsoft SQL Server 2005 Express will be installed on your machine first, then you will have the option to create a new Application Catalog database on that server that is populated with sample data.
- **If you do have access to a database server**, but just do not have any sample data in it, select the **Create a new Application Catalog** option, and choose to populate this new Application Catalog with sample data by selecting the **Include Sample Data in New Catalog** option on the **Create Application Catalog** dialog box.

Connecting to an Existing Application Catalog



Edition • AdminStudio Enterprise Server Tools are included with AdminStudio Enterprise Edition.

From AdminStudio and the AdminStudio client tools, you usually connect to an Application Catalog by selecting **Connect** on the **Catalog** menu. AdminStudio supports SQL Server databases.



Note • Starting with AdminStudio 9.01, Oracle databases are not supported.

You can choose to connect to a standalone Application Catalog database or the AdminStudio Enterprise Server Application Catalog database.

- [Connecting AdminStudio Client Tools to a Standalone Application Catalog](#)
- [Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog](#)



Note • See [Standalone Application Catalog vs. the AdminStudio Enterprise Server Application Catalog](#) for more information.



Note • AdminStudio Enterprise Tools are always connected to the AdminStudio Enterprise Server database.

Connecting AdminStudio Client Tools to a Standalone Application Catalog

To connect to an existing Standalone Application Catalog from an AdminStudio client tool (Application Manager, ConflictSolver, or PackageExpert), perform the following steps.



Task: *To connect to an existing Standalone Application Catalog from the AdminStudio client tools:*

1. Perform one of the following:
 - **Application Manager and ConflictSolver**—On the **Catalog** menu, click **Connect**.
 - **PackageExpert**—Select the **Application Catalog** node and select **Connect** from the context menu.

The **Connect Application Catalog** dialog box opens, displaying three tabs: **Enterprise Server**, **Standalone**, and **Recent**.
2. Click the **Standalone** tab. The Standalone tab opens, prompting you to enter database connection information.
3. If you want this Application Catalog to be the default shared Application Catalog used in your organization, select the corresponding option at the bottom of the dialog box.
4. Select the **Server** where the Application Catalog is stored.
5. Specify how the database server should verify the authenticity of the login—either using **Windows Authentication** or **Server Authentication**. If you selected **Server Authentication**, enter the appropriate **Login ID** and **Password**.
6. In the **Catalog** box, enter the name of the Application Catalog you want to open.
7. Click **Test** to test the connection to the database.
8. Click **OK**.

Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog



Edition • AdminStudio Enterprise Server Tools are included with AdminStudio Enterprise Edition.

The AdminStudio Enterprise tools (Report Center, Job Manager, and Workflow Manager) are configured during installation to connect to an Application Catalog, which is referred to as the Enterprise Server Application Catalog.

You can also connect the AdminStudio *client* tools to the Enterprise Server Application Catalog. This allows you to have all of the client and enterprise tools reference the same database.

To connect to the AdminStudio Enterprise Server application catalog from an AdminStudio client tool, perform the following steps.



Task:

To connect to the AdminStudio Enterprise Server Application Catalog:

1. Perform one of the following:
 - **Application Manager and ConflictSolver**—On the **Catalog** menu, click **Connect**.
 - **PackageExpert**—Select the **Application Catalog** node and select **Connect** from the context menu.

The **Connect Application Catalog** dialog box opens, displaying three tabs: **Enterprise Server**, **Standalone**, and **Recent**.

2. Open the **Enterprise Server** tab.
3. The URL to the AdminStudio Enterprise Server is listed above the **Authentication** field. If the AdminStudio Enterprise Server has not yet been configured with the AdminStudio client tools (such as when it is set to its default value of `http://localhost`), click the URL link to open the **Select AdminStudio Enterprise Server URL** dialog box, and enter the URL for location of the AdminStudio Enterprise Server associated with this installation of AdminStudio.
4. From the **Authentication** list, select either **AdminStudio Enterprise Server User** or **Windows Authentication**.



Important • When using **AdminStudio Enterprise Server User** authentication, if *Anonymous authentication is turned off in IIS*, both the user's machine and the AdminStudio Enterprise Server need to be on the same domain in order for login to succeed.

5. If you selected **AdminStudio Enterprise Server User**, enter your AdminStudio Enterprise Server **User Name** and **Password** (provided by your System Administrator).
6. Click **Login**. After a successful login, the **Provider**, **Server**, and **Catalog** name of the Enterprise Server database is listed.
7. Click **OK**.

Login Troubleshooting

If you are using a Web Portal with custom security zone settings, your AdminStudio Enterprise Server URL is using an IP address, and you receive Error 0x800A1518 when you attempt to login, change the AdminStudio Enterprise Server URL to the NetBios equivalent and then try again. For example, if you are connecting to `http://120.12.1.15`, the NetBios equivalent would be `http://wfmporatl`.

Creating New Application Catalogs

You can create new Application Catalogs using the AdminStudio interface, or using scripts. This section includes the following topics:

- [Creating New Application Catalogs Using the AdminStudio Interface](#)
- [Creating New Application Catalogs Using Scripts](#)

Creating New Application Catalogs Using the AdminStudio Interface

To create a new SQL Server application catalog database, perform the following steps.



Task: *To create a new standalone Application Catalog:*

1. On the AdminStudio **Catalog** menu, click **Create**. The **Welcome** panel of the **Application Catalog Wizard** opens.
2. If you want to include a few sample packages in this new Application Catalog, select the **Include Sample Data in New Catalog** option.
3. Click **Next**. The **Specify Database Information** panel opens.
4. Enter or select the name of the **Server** where this Application Catalog will be stored.
5. Specify how the database server should verify the authenticity of the login—either using **Windows Authentication** or **Server Authentication**. If you selected **Server Authentication**, enter the appropriate **Login ID** and **Password**.
6. In the **Catalog** box, enter the name of the Application Catalog you are creating.
7. If you are creating a new SQL Server database which is intended for use as a Subscriber for Application Catalog Replication, select the **Create catalog as a subscriber** option. If you do not intend to use the Replication functionality that AdminStudio provides for SQL Server, do not select this option. For more information, see [About Creating a Catalog as a Subscriber](#) and [Replicating Application Catalogs](#).
8. Click **Test** to test the connection to the database.
9. Click **Next**. The **Select Software Repository Location** panel opens, prompting you to select the location where the Software Repository will store imported packages and their associated files. For more information, see [Using the Software Repository](#).

10. If you want to store data associated with this Application Catalog's packages in the Software Repository, choose the **Enable the Software Repository** option, select a Software Repository Location, and enter a Login ID and Password of the **Proxy Account** that must be used to access this Repository.
11. Click Next. The **Creating Application Catalog** panel opens and reports on the creation progress. When the Application Catalog has been created, a message appears stating that the creation was successful.
12. Click Finish. The new Application Catalog now opens in AdminStudio.

About Creating a Catalog as a Subscriber

Please note the following about creating a Subscriber Application Catalog for Application Catalog Replication:

- You cannot create a Publication in a Subscriber database.
- If you *choose* the **Create catalog as a subscriber** option, making it a **Subscriber** database, the **Subscription Manager** option under **Replication** on the **Catalog** menu is enabled.
- If you *do not choose* this option, making it a **Publisher** database, the **Publication Manager** option under **Replication** on the **Catalog** menu is enabled.
- Once you have created a database, you cannot go back and edit the choice you made on this option.
- Just because you select this option does not mean that this database must contain subscribed data or that all of its data must be subscribed. It can contain all local data or both local and subscribed data.

For more information, see [Replicating Application Catalogs](#).



Tip • The first time AdminStudio is run, anyone can create an SQL Server database (providing they have access to an SQL Server and database creation rights). However, once an AdminStudio Application catalog has been created, the database creator becomes the Application Catalog administrator, and security rights are in place in AdminStudio.

Creating New Application Catalogs Using Scripts

Typically, users with administrative privileges in AdminStudio use **Create** on the **Catalog** menu to create a new Application Catalog.

However, because of security concerns, some Database Administrators may be hesitant to grant the database creation rights that are necessary to create an Application Catalog database using SQL Server to AdminStudio users. Consequently, the Database Administrator must manually create the database using scripts and provide the necessary read and write access for users to that database.

AdminStudio is shipped with database creation SQL scripts to make it easy for Database Administrators to create new Application Catalogs:

Table 6-3 • Scripts to Create Application Catalog Databases on SQL Server

Script File	Description
AMS_System_Schema.SQL	Primary Workflow Manager tables
AMSCreateIndex.SQL	Indexes for Workflow Manager tables
AMS_SamplePackagingTemplate.SQL	Sample Workflow Manager template data
AS_System_Schema.SQL	Primary AdminStudio client tools data
Seed_Data.SQL	Seed data required by AdminStudio
CustomReportWizard.SQL	Report Center data

These SQL scripts are located in the following directory:

AdminStudio Installation Directory\Support\SQL_Scripts

To create an Application Catalog database on SQL Server, perform the following steps:



Task: *To use scripts to create an AdminStudio Application Catalog on SQL Server*

1. Log on to your SQL Server.
2. Launch the Enterprise Manager and Query Analyzer.
3. In Query Analyzer, execute a CREATE DATABASE command to create and identify the new Application Catalog database.
4. Select the newly created database in Query Analyzer.
5. Execute the following scripts in order:
 - a. AMS_System_Schema.SQL
 - b. AMSCreateIndex.SQL
 - c. AMS_SamplePackagingTemplate.SQL
 - d. AS_System_Schema.SQL
 - e. Seed_Data.SQL
 - f. CustomReportWizard.SQL

Specifying a Default AdminStudio Application Catalog

You can specify a default Application Catalog so that each time you open AdminStudio, you will be prompted to login to the same Application Catalog database.

You can also configure your enterprise so that *all of the users at your enterprise* will be prompted to login to the same Application Catalog each time they open AdminStudio.

Setting a Default Application Catalog for Yourself

Whenever you connect to an Application Catalog, you can designate it as the default Application Catalog by selecting the **Make this the shared default Application Catalog** option on the **Connect Application Catalog** dialog box.

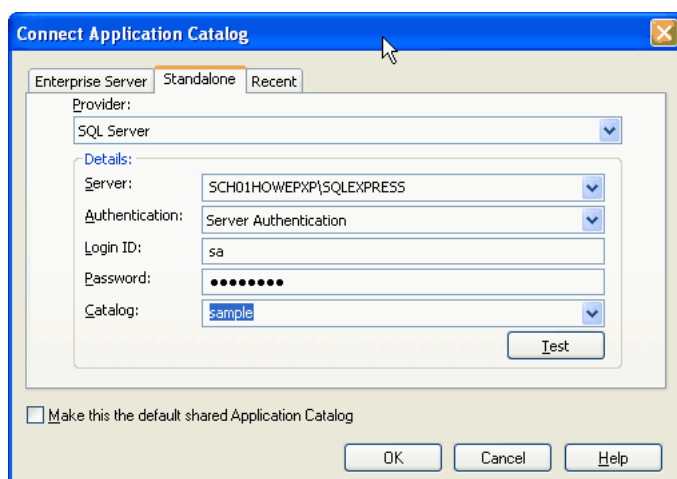


Figure 6-2: Default Shared Application Catalog Option on the Connect Application Catalog Dialog Box

Setting the Default Application Catalog for Your Enterprise

To configure your enterprise so that *all of the users at your enterprise* will be prompted to login to the same Application Catalog each time they open AdminStudio, all users need to be connected to the same AdminStudio Shared.ini file that is stored in a shared network location that is available to all users.

Methods to Set the Default Application Catalog for Your Enterprise

You can set a default Application Catalog for all AdminStudio users in your enterprise in two ways:

- Using the AdminStudio Interface
- or
- Editing the Shared AdminStudio.ini File

Using the AdminStudio Interface

To use the AdminStudio interface to set a default Application Catalog for all AdminStudio users at your enterprise, perform the following steps.



Task: *To specify the default Application Catalog:*

1. First, the AdminStudio System Administrator needs to perform the following steps to set the default Application Catalog for the enterprise:
 - a. Locate and copy the following file on the machine where you installed AdminStudio:
`C:\AdminStudio Shared\Shared AdminStudio.ini`
 - b. Copy this file to a shared network location that is accessible to all of the users in your enterprise.
 - c. Launch AdminStudio.
 - d. On the **Tools** menu, click **Options**. The AdminStudio **Options** dialog box opens.
 - a. Open the **Locations** tab.
 - b. Set the **AdminStudio Shared Location** to the network location where you copied the Shared AdminStudio.ini file, such as:
`\\servername\AdminStudio Shared`
 - c. Open the **Application Catalog** tab.
 - d. Click **Change** next to the **Shared AdminStudio Application Catalog** field. The **Select Application Catalog** dialog box opens.
 - e. Select the Application Catalog database that you want to be set as the default, following the instructions in [Connecting to an Existing Application Catalog](#).

After you connect to the database, that database name is listed in the **Shared AdminStudio Application Catalog** field.
 - f. Click **OK** to close the **Options** dialog box.
2. Next, each AdminStudio user in the enterprise needs to perform the following steps to set the location of *their* **AdminStudio Shared** directory to the same shared network directory that the System Administrator configured.
 - a. Launch AdminStudio.
 - b. On the **Tools** menu, click **Options**. The AdminStudio **Options** dialog box opens.
 - a. Open the **Locations** tab.
 - b. Set the **AdminStudio Shared Location** to the shared network location provided by your System Administrator.



Caution • If a user is not assigned to a Role that has the **Modify AdminStudio Tools Options Dialog** permission, they cannot change the location of the **AdminStudio Shared** directory on the **Options** dialog. In this situation, the location of the **AdminStudio Shared** directory would be set during installation.

Editing the Shared AdminStudio.ini File

To set a default Application Catalog for all AdminStudio users at your enterprise by editing the Shared AdminStudio.ini file, perform the following steps.



Task: *To specify the default Application Catalog by editing the Shared AdminStudio.ini file:*

1. First, the AdminStudio System Administrator needs to perform the following steps to set the default Application Catalog for the enterprise:
 - a. Locate and copy the following file on the machine where you installed AdminStudio:
`C:\AdminStudio Shared\Shared AdminStudio.ini`
 - b. Copy this file to a shared network location that is accessible to all of the users in your enterprise.
 - c. Open the Shared AdminStudio.ini file that you just copied to a shared network location.
2. Insert the following in the [Database Settings] section of the Shared AdminStudio.ini file:

```
[Database Settings]
DefaultDatabase=Provider=SQLOLEDB.1;User ID=userid; PWD=password;Initial
Catalog=nameofdatabase;Data Source=nameofsqlserver;
```
3. Next, each AdminStudio user in the enterprise needs to perform the following steps to set the location of *their* **AdminStudio Shared Location** directory to the same shared network directory that the System Administrator configured.
 - a. Launch AdminStudio.
 - b. On the **Tools** menu, click **Options**. The AdminStudio **Options** dialog box opens.
 - a. Open the **Locations** tab.
 - b. Set the **AdminStudio Shared Location** to the shared network location provided by your System Administrator.



Caution • If a user is not assigned to a Role that has the **Modify AdminStudio Tools Options Dialog** permission, they cannot change the location of the **AdminStudio Shared** folder on the **Options** dialog. In this situation, the location of the AdminStudio Shared location would be set during installation.



Note • The Roles assigned to a user determine that user's permissions:

- The **Create** and **Connect** options on the **Catalog** menu on the AdminStudio Interface are disabled for users that are not assigned to a Role that has permission to perform those actions.
- Users that are assigned to Roles that have the **Modify AdminStudio Tools Options Dialog** permission can change the location of the **AdminStudio Shared** folder on the **Options** dialog (accessed by selecting **Options** on the **Tools** menu from the AdminStudio Interface). For those users who do not have that permission, **Options** on the **Tools** menu is disabled, so they are unable to change the location of the **AdminStudio Shared** folder.

Searching an Application Catalog

You can search for data in Application Catalog tables by using **Find** on the **Edit** menu. You can search all tables in all packages in a group, or just search one column in one table of one package.



Note • This search is limited to string type columns.

The tables that are searched depend upon what is selected when the **Find** dialog box is opened:


Table 6-4 • Application Catalog Search Options

If you select...	and specify these options...	this will be searched
Application Catalog	All Tables and All Columns	All tables and all columns in all of the packages in the Application Catalog.
Group	All Tables and All Columns	All tables and all columns in all of the packages in the selected group.
Package	All Tables and All Columns	All tables and all columns in the selected package.
Package	A table and All Columns	All columns of a specific table in the selected package.
Package	A table and a column	A specific column in a specific table in the selected package.



Task: **To search the Application Catalog:**

1. Open an Application Catalog in Application Manager or ConflictSolver.
2. Select the node in the tree (Application Catalog, Groups, a specific group, a specific package, an OS snapshot, etc.) that you want to search.
3. On the **Edit** menu, click **Find**. The **Find** dialog box opens.

You could also click the Find button () on the toolbar, use Ctrl + F, or choose Find from the selected package or group's context menu.

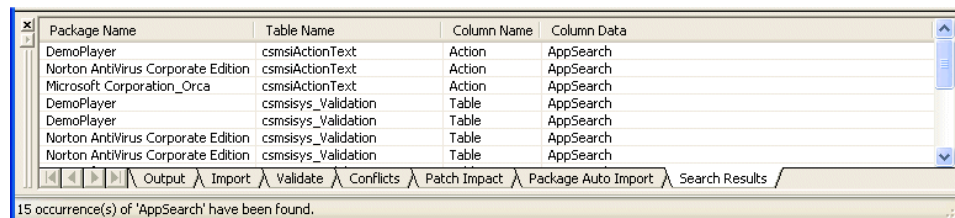
4. In the **Find What** text box, enter the text that you want to search for.



Note • This search is limited to string type columns.

5. On the **Look In Table** list, select the table that you would like to search, or select **<All Tables>**. When you select a table from this list, the **Look In Columns** list is populated with all of the columns in that table.
6. If you selected a table from the **Look in Table** list, all of the columns in that table are listed. Select the column that you would like to search, or select **<All Columns>**.
7. If you want to search for a partial match rather than an exact match, select the **Partial Match** option.
 - **If this option is not selected**, Application Manager will search for an exact match of the text you entered in the **Find What** text box. The search will be case sensitive.
 - **If this option is selected**, then Application Manager will use appropriate wild card characters so that a partial data match is performed. The search will be case insensitive
8. Click **Find** to initiate the search.

The **Find** dialog box will close, and the data that is found is displayed in the **Search Results** tab of the Output Window, in the following format:



Package Name	Table Name	Column Name	Column Data
DemoPlayer	csmsiActionText	Action	AppSearch
Norton AntiVirus Corporate Edition	csmsiActionText	Action	AppSearch
Microsoft Corporation_Orca	csmsiActionText	Action	AppSearch
DemoPlayer	csmsisys_Validation	Table	AppSearch
DemoPlayer	csmsisys_Validation	Table	AppSearch
Norton AntiVirus Corporate Edition	csmsisys_Validation	Table	AppSearch
Norton AntiVirus Corporate Edition	csmsisys_Validation	Table	AppSearch

15 occurrence(s) of 'AppSearch' have been found.

If you double click on the data displayed in the Output Window, Application Manager will navigate to the appropriate record in the **Tables View**, and that record will be highlighted in red.

Disconnecting from an Application Catalog

To disconnect from the currently open Application Catalog, select **Disconnect** from the **Catalog** menu. You can disconnect from an Application Catalog while you are in ConflictSolver, Application Manager, PackageExpert, and also when you are in the AdminStudio Interface.

When you have disconnected from an Application Catalog in ConflictSolver or Application Manager, the following message appears:

ConflictSolver requires a connection to the AdminStudio Application Catalog. Select Connect from the Catalog menu to establish a connection.

Upgrading an Existing Application Catalog

When you attempt to open an AdminStudio 5.0, 5.5, 6.0, 7.0, or 7.5 Application Catalog in AdminStudio 9.5, you are prompted to upgrade it to use the AdminStudio 9.5 schema.

Log files for the upgrade are created in the following directory:

AdminStudio Shared Directory\ConflictSolver\Logs



Note • Note the following regarding upgrading an existing Application Catalog:

- The upgrade of AdminStudio 3.0, 3.01, and 3.5 databases is not supported by AdminStudio 7.0 or later.
- Starting with AdminStudio 8.0, Microsoft Access databases are not supported.
- Starting with AdminStudio 9.01, Oracle databases are not supported.
- When an SQL Server Application Catalog database is upgraded, the old tables are not dropped from the Application Catalog.

Upgrading Pre-AdminStudio 5.0 Application Catalogs

Pre-AdminStudio 5.0 Application Catalogs cannot be upgraded automatically by AdminStudio 7.0 or later. However, you can upgrade them using the Legacy Upgrade Wizard, a standalone utility that was included with AdminStudio 7.0 and 7.5. The Legacy Upgrade Wizard utility is installed in the following directory:

C:\Program Files\InstallShield\AdminStudio\7.x\Common\LegacyUpgradeWizard.exe

If you do not have a copy of AdminStudio 7.0 or 7.5 available to you, contact Technical Support.

Organizing Your Application Catalog Using Groups

Within Application Manager and ConflictSolver, you can create groups to organize your products, patches, and OS Snapshot images in the Application Catalog. This is especially useful for organizing your Application Catalog in ways consistent with how your company is organized.

For example, you could create a group representing a certain department's base image including the proper operating system and necessary applications. When you perform conflict analysis on new packages you are integrating into your environment, you can run only the relevant comparisons—saving you the time it would take to run the analysis against all packages in the Application Catalog, or the effort of manually determining the set of packages against which you want to run the analysis each time.

Tasks relating to groups include:

- [Adding Groups](#)
- [Organizing Products in Application Manager](#)
- [Deleting Application Manager Groups](#)
- [Editing Group Properties](#)
- [Copying Packages to Multiple Groups](#)
- [Moving Products, OS Snapshots, and Groups](#)

Adding Groups

To add additional groups or subgroups to an Application Catalog, perform the following steps.



Task: *To add a group to Application Manager:*

1. In the **Application Manager Product View**, right-click on the group to which the new group should belong and select **New Group**.
2. Provide a name for the new group.
3. Press **Enter**.

Organizing Products in Application Manager

To move products or groups into different groups, perform the following steps:



Task: *To organize individual packages and groups of packages in Application Manager:*

1. From the **Application Manager Product View**, select a product or group you want to move.
2. Drag the product or group onto a new group.



Note • *The following rules apply to drag and drop operations in the Application Manager Product View:*

- *You cannot drop a node on itself.*
- *You cannot drop a node on its parent. It is already a child of the parent.*
- *You cannot drop a group on its child groups.*

Deleting Application Manager Groups

To delete a group from an Application Catalog, perform the following steps:



Task: *To delete a Application Manager group:*

1. Right-click the group you want to delete in the **Application Manager Product View** and select **Delete** from the context menu.
2. From the resulting message box, confirm the deletion.



Caution • When you delete a group, all subgroups and packages within that group are also removed from the Application Manager.

Editing Group Properties

For each group, you can modify its name, and add a description or other comments. To edit this information, perform the following steps.



Task: *To edit group properties:*

1. Right-click on the group from the **Application Manager Product View** and select **Properties** from the context menu. The **Group Properties** dialog box opens, displaying the group **Name**, **Description**, and **Comments**.
2. Make any desired edits.
3. Click **OK**.

Copying Packages to Multiple Groups

More than one copy of the same package can exist in an Application Catalog. You can copy a package from one group into another group.



Task: *To copy a package to multiple groups:*

1. Select the package from the **Application Manager Product View**.
2. While holding down the **Ctrl** key, drag the package to the group into which you want it copied.



Note • You can also right-click on a package and use the copy and paste commands to copy the item.

Moving Products, OS Snapshots, and Groups

In addition to copying OS Snapshots or Products to multiple groups, you can also move OS Snapshots, Products, or Groups into other groups.



Task: *To move a product, OS Snapshot, or group:*

1. Select the product, OS Snapshot, or group you want to move in the **Application Manager Product View**.
2. Drag the item into a new group.



Note • You can also right-click on a product, OS Snapshot, or group and use the cut, copy, and paste commands to move or copy the item.

Viewing and Editing Package Extended Attributes

Extended attributes are optional attributes for packages, defined by an [Extended Attribute Description File](#) (in XML format). Because you can manually create the description file, you have the flexibility to include information about each package that may be specific to your organization—such as the users or business groups that receive the package.

Extended attributes can be edited in the **Extended Attributes** view on a package-by-package basis.

The following tasks and concepts relate to extended attributes:

- [Using Extended Attributes](#)
- [Extended Attribute Description File](#)
- [Integrating Package Extended Attribute Data with an Application Request](#)



Note • You can also edit a package's extended attributes in the ConflictSolver Process Assistant. See [Extended Attributes Page](#).

Using Extended Attributes

Assuming you have created an extended attribute description file (or are using the default provided file), you can configure Application Manager to use it with the current Application Catalog.



Task: *To use extended attributes in Application Manager:*

1. With Application Manager open, click **Options** on the **Tools** menu. The ConflictSolver and Application Manager **Options Dialog Box** opens.
2. Click the **Extended Attributes** tab.
3. Specify or browse to the extended attribute description file (.xml) containing the extended attributes you want to use for Application Manager.
4. Click **OK**.

The **Extended Attributes View** is available under each package in Application Manager.



Caution • The default Extended Attribute description file is named *EA_Default.xml*, and is installed in the AdminStudio Shared folder. You can modify the data displayed in the **Extended Attributes** view, but to do this, do not edit the *EA_Default.xml* file. Instead, copy the *EA_Default.xml* file, rename it, make your edits to the new file, and then enter the new file name and location in the Extended Attribute Description File field on the **Extended Attributes** tab of the ConflictSolver and Application Manager **Options** dialog box.

Extended Attribute Description File

Application Manager uses an XML file to describe the data that appears in the **Extended Attributes** view. The name and location of this XML file can be specified from the **Extended Attributes** tab in the ConflictSolver and Application Manager **Options** dialog box.



Caution • The default Extended Attribute description file is named *EA_Default.xml*, and is installed in the AdminStudio Shared folder. You can modify the data displayed in the **Extended Attributes** view, but to do this, do not edit the *EA_Default.xml* file. Instead, copy the *EA_Default.xml* file, rename it, make your edits to the new file, and then enter the new file name and location in the Extended Attribute Description File field on the **Extended Attributes** tab of the ConflictSolver and Application Manager **Options** dialog box.

Description File Properties

The description file, which is in XML format, contains tags for each extended attribute (up to a limit of 400 attributes). It supports text or file values. The following list explains each tag available in the description file.

Table 6-5 • Description File Tags

Attribute	Description
UniqueIdentifier	This value, which Application Manager uses to validate that the XML file is for extended attributes, must be set to ISASEA40.
Name	The name of the attribute as it appears in the Extended Attributes view. This cannot exceed 255 characters.
Type	The extended attribute type. This can be Text, File, or Selection. If no type is specified, then Application Manager defaults the attribute to text.
DefaultValue	This tag, available only for Text types, provides the default value for the attribute. This optional value cannot exceed 512 characters.
DefaultFileExtension	This tag, available only for File types, provides the default file extension when you browse for the file. Examples of this could be *.txt, *.bmp, *.doc, or *.* (representing all files).
FileFilter	<p>Provide the file types to populate the File type filter in the Browse dialog box. These must be in pairs, and in the format Longname (*.ext) *.ext. Before the closing </FileFilter> tag, you must have two pipe symbols ().</p> <p>For example, to include filters for text files, bitmaps, and all files, use the following line:</p> <pre><FileFilter>Text Document (*.txt) *.txt Bitmap (*.bmp) *.bmp All Files (*.*) *.* </FileFilter></pre> <p>This value cannot exceed 255 characters.</p>
Caption	The caption for the Browse dialog box when using File types. This cannot exceed 255 characters.
Values	Used only for Selection types, this is a semicolon-delimited list of possible values for the selection. These will appear in a drop-down list for the extended attribute. The first value is used as the default. The total number of characters of all the values and necessary semicolons cannot exceed 255 characters.
HelpText	Text that appears below the value field for either Text or Selection attributes. You can use it to provide additional information to help users know what to input. This cannot exceed 512 characters.

Description File Format

An example of an extended attribute description file follows:

```
<Extended_Attribute UniqueIdentifier="ISASEA40">
  <AttributeDetails>
    <Name>Owner</Name>
    <Type>Text</Type>
    <DefaultValue></DefaultValue>
    <HelpText>Provide the name of the package's owner.
    </HelpText>
  </AttributeDetails>
  <AttributeDetails>
    <Name>Test Script</Name>
    <Type>File</Type>
    <DefaultFileExtension>*. *</DefaultFileExtension>
    <FileFilter>All Files (*.*)|*.*||</FileFilter>
    <Caption>Test Script Files</Caption>
  </AttributeDetails>
  <AttributeDetails>
    <Name>Program Type</Name>
    <Type>Selection</Type>
    <Values>Office Application;Utility;
    Graphic Application;Programming Application;Game;Other
    </Values>
    <HelpText>Select the type of application from the
    above list.
    </HelpText>
  </AttributeDetails>
</Extended_Attribute>
```

Integrating Package Extended Attribute Data with an Application Request



Note • AdminStudio Workflow Manager is a Web-based application management system that has integrated functionality with AdminStudio.

Application Manager allows you to integrate extended attributes with AdminStudio Workflow Manager. This option is enabled by selecting the **Integrate with Workflow Manager** option on the **Extended Attributes** of the ConflictSolver and Application Manager **Options** dialog box.

When the **Integrate with Workflow Manager** option is selected, you can associate extended attribute data for packages in Application Manager with Application Requests in Workflow Manager. This is accomplished by right-clicking on the package name in the Application Manager Product View and selecting **Associate with Workflow Manager Application** from the context menu.



Task: *To associate a package with a Workflow Manager Application Request:*

1. Open Application Manager.
2. Connect to the AdminStudio Enterprise Server Application Catalog. See [Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog](#).
3. On the **Tools** menu, click **Options**. The **Options** dialog box opens.
4. On the **Extended Attributes** tab, confirm that the **Integrate with Workflow Manager** option is selected.
5. In the Application Manager **Product View**, select the product that you want to link to Workflow Manager and select **Associate with Workflow Manager Application** from the context menu.

The **Associate with Workflow Manager Application** dialog box opens.

6. Pick the application in Workflow Manager with which you want to associate this product.

Assuming that Workflow Manager is configured to use the same extended attributes file, if you enter new data into the Extended Attributes view in Application Manager, Workflow Manager automatically detects it; if changes are made in Workflow Manager, they are automatically reflected in Application Manager.

By design, extended attributes data in Application Manager and Workflow Manager data have a one-to-one relationship. You can only associate one Workflow Manager application with a product in Application Manager; once the application is associated, it is no longer available for association with other Application Manager products.



Note • Another integration feature between AdminStudio and Workflow Manager is that when you associate a package with a Workflow Manager Application Request and then view that Application Request's Application Report, there is a link to open the Package Report of its associated package. There is also a link on the Package Report to open the Application Report of its associated Application Request.

Viewing Package History


The tracking of change history is a critical operation within the Enterprise environment. Maintaining this information, displaying it, and allowing it to be a filter will give you the information you need to monitor and maintain the integrity of your software packages.

In AdminStudio, any operation that materially changes a software package or the data associated with the package is tracked, and can be viewed in the Application Manager **History** view.



Task: *To view a package's history:*

1. Open **Application Manager**.
2. In the **Application Manager Product View**, expand a package's tree and select the **History** node. The **History View** opens, displaying the following information:

Item	Description
Maximum History Log Per Package	Enter the maximum number of History Log entries you want to store for each package. When the number of log entries exceeds this amount, the oldest entry will be deleted.
Action	<p>Name of the event which was logged:</p> <ul style="list-style-type: none"> • Import/Reimport • Validation • Conflict Detection • Conflict Resolution • Extended Attribute Modification • Package Description Modification • Package Move/Copy • Patch Impact Analysis  <p>Note • You can specify which events you want to be listed on the History View by making selections on the History Tab of the ConflictSolver and Application Manager Options Dialog Box. You can choose which events you want to log and which events you want to display in the History View.</p>
Date	Date and time logged event occurred.
User	User who performed the logged event.
Description	Description providing details of the logged event.



Note • If a package was replicated into another Application Catalog, its history data would not be replicated.

Deleting Package History

To delete all of the entries in a package's History Log File, perform the following steps:



Task: *To delete package history:*

1. Open **Application Manager**.
2. Select the package and right-click to open the context menu.
3. On the context menu, point to **Delete** and click **History Log Information**.

Deleting Packages

You can delete packages that have been imported into the Application Catalog.



Task: *To delete a package from Application Manager:*

1. Right-click on the package in the **Application Manager Product View**, point to **Delete** on the context menu, and click **Package** (to delete the package from the selected Group) or **Package from all Groups** (to delete the package from all Groups in the Application Catalog).
2. Confirm the deletion.

Importing Data

You can use the Import Wizard to import the following types of data into the Application Catalog:

Importing Windows Installer Packages, Transforms, and Patches

You use the Application Manager Import Wizard to import Windows Installer packages (.msi) with any associated transforms (.mst) and patches (.msp) into the Application Catalog. See [Importing Windows Installer Packages, Transforms, and Patches](#).

Ad-Hoc Import

You can also import transform files and patch files after the package that they are associated with has already been imported into the Application Catalog. This is referred to as an ad-hoc import. The imported transform or patch file is then stored in the Software Repository along with the rest of the imported package's associated files. See [Using the Software Repository](#) and [Performing an Ad-Hoc Import of Transform Files or Patch Files](#).



Important • You can only perform an ad-hoc import if you are connected to a Software Repository-enabled Application Catalog.

Importing Virtual Packages

You can import Microsoft App-V, VMware ThinApp, and Citrix virtual packages into the Application Catalog. You have several options in how to import a virtual package:

- **During the import of the Windows Installer package**—During the import of a Windows Installer package, if the Import Wizard searches for and finds a virtual package created by AdminStudio, you are prompted to also import that virtual package. See [Importing a Virtual Package During the Import of its Source Windows Installer Package](#).
- **After the import of the Windows Installer package**—If you import an AdminStudio-created virtual package into an Application Catalog that already contains the virtual package's source Windows Installer package, you are prompted to associate the virtual package with the Windows Installer package. See [Importing a Virtual Package After the Import of its Source Windows Installer Package](#).
- **By itself**—You can use the Import Wizard to import a virtual package directly into the Application Catalog. Then you can use the **Associate Package** function to associate the virtual package with its source package. See [Importing a Virtual Package Without Its Source Windows Installer Package](#) and [Manually Associating a Virtual Package with a Windows Installer Package](#).

Importing Merge Modules

A merge module (.msm) is a package containing all of the logic and files needed to install distinct pieces of application functionality such as run-time .dll files and virtual machines. Merge modules are built once and can be added to any installation project. Merge modules, which are imported into the Application Catalog using the Import Wizard, are used primarily during conflict detection.

For optimal performance, Merge modules should be imported into an Application Catalog database prior to importing Windows Installer packages. This ensures that conflicts resulting from not using available merge modules are correctly identified. For more information, see [Importing Merge Modules](#).

Additional Data

Additionally, you can use the Import Wizard to import OS snapshots (.osc), Marimba Native Channel Packager (.ncp) files and other non-MSI based setup programs. See [Importing OS Snapshots](#), [Importing Marimba NCP Files](#), and [Importing Other Setup Types](#).

Importing Windows Installer Packages, Transforms, and Patches

You can import a Windows Installer package with all of its associated transform files and patches into the Application Catalog at the same time. The transform files and patches are immediately associated with the Windows Installer package.

Tasks associated with importing Windows Installer packages, transforms, and patches include:

- [Importing Windows Installer Packages](#)
- [Using Duplicate Package Identifiers](#)
- [Performing an Ad-Hoc Import of Transform Files or Patch Files](#)

Once you import these files, they become available for conflict identification.



Note • Although you can identify conflicts between an external package and packages already in the Application Catalog, it is strongly recommended you import the package first and perform conflict identification after import. Conflict information is only persisted for the last conflict identification performed between an external package and packages in the Application Catalog, and performance is better when performing conflict identification between internal packages. See [Conflict Persistence](#) for more information.

Importing Windows Installer Packages

You can use the Import Wizard to import Windows Installer packages with all of its associated transform and patch files.

Application Manager stores imported Windows Installer (.msi) packages for further conflict analysis. When you import a package into an Application Catalog, you can automatically run validation and conflict identification features, if configured through the ConflictSolver and Application Manager **Options** dialog box.




Task:


To import a Windows Installer package with any associated Transform and Patch files into an Application Catalog:

1. On the **Catalog** menu, click **Import Package**. The **Import Wizard Welcome** panel opens.
2. Click **Next**. The **File Selection** panel opens.
3. Click **Browse** and select the Windows Installer package that you want to import.
4. If you want to add the selected package to the Software Repository, select **Add the file(s) to the Software Repository** option. After you select this option, if AdminStudio determines that there are other files in that directory that could be associated with the selected file, the **Additional Dependencies of the Import File** list appears, enabling you to select any additional files to be managed by the Software Repository.
5. When you are finished making selections on the **File Selection** panel, click **Next**. The **MST Source Information** panel opens, where you may optionally select any additional transform (.mst) files to be imported along with the Windows Installer package.



Note • When the **MST Source Information** panel opens, all of the .mst files that are in the same directory as the Windows Installer file you are importing are automatically listed in the **Transform Files (.mst)** list, but only those .mst files that AdminStudio determines are probably applicable to this Windows Installer package are selected to be included in the import. If you do not want to import a selected .mst file, clear the selection.

6. Should the package require transforms, click the **New** button () in the **Transforms** area to browse to the location of the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the **Move Up** and **Move Down** buttons. If you need to delete a transform you have added, select the transform and click the **Delete** button.

7. When you have finished adding transforms, click **Next**. The **MSP Source Information** panel opens, where you may optionally select any additional patch (.msp or .exe) files to be imported along with the Windows Installer package.
8. Should the package require patches, click the **New** button () in the Patches area to browse to the location of the patch. If the package requires multiple patches, you can repeat the procedure as necessary. The order in which patches are applied can be changed by selecting a patch and clicking the **Move Up** and **Move Down** buttons. If you need to delete a patch you have added, select the patch and click the **Delete** button.
9. When you have finished adding patches, click **Next**. If the package that you are importing includes some custom tables, the **Import Options** panel opens, prompting you to specify whether you want to import these tables.
10. To add one of the listed tables to the **Ignore Tables** list on the **Import** tab of the ConflictSolver and Application Manager **Options** dialog box (so that it will not be imported into the Application Catalog during any import), select the table name and click **Add to Ignore List**.
11. Click **Next**. The **Destination Group** panel opens.



Note • If you opened the Import Wizard by selecting a Group in the Application Catalog and then selecting **Import Package** from the context menu, the package will be imported into the selected group, and this panel will not appear.

12. Select one or more destination groups into which your package will be imported. You can also click **New** to create a new Group. When you click **Next**, the **Summary** panel opens.
13. Review the information in the **Summary** panel. If you are satisfied with the import options, click **Finish** to start the import.

A report of the import process appears on the **Import** tab of the **Output** window. Depending on whether options have been set from the **General** tab of the ConflictSolver and Application Manager **Options** dialog box (available from the **Tools** menu), validation and conflict checking may be performed during import.

The Windows Installer package appears in the **Application Manager Product View** under the selected group.

About Importing Windows Installer Packages

Regarding importing Windows Installer packages, note the following:

- If you specify an update.exe patch file that was created by Developer/DevStudio/InstallShield Editor, Application Manager will extract the .msp file in the Temp folder and then perform the import.
- For patches to be applied to an MSI package, it is necessary to perform an Administrative install of the MSI package and then perform an Administrative install of each patch package one by one. This way, the content of each patch package is appended to the MSI package at the Administrative install location.
- You can also import a Windows Installer package by right-clicking the group into which you want to import the package and selecting **Import Package** from the context menu. The **MSI Source Information** panel is displayed in the Import Wizard, and you can proceed with the steps above. However, since you have already

specified the group into which you want to import the package, the **Destination Group** panel does not appear.

- Although not required, it is recommended that you import only Windows Installer packages that have passed the Full MSI Validation Suite validation into the Application Catalog.

About Windows Installer Packages (.msi)

Application Manager supports the import of Windows Installer packages (.msi). A Windows Installer package contains all of the information that the Windows Installer requires to install or uninstall an application or product and to run the setup user interface. The .msi file can also contain one or more transform files (.mst) and one or more patches (.msp).

A Windows Installer package is organized around the concepts of components and features:

- A feature is a part of the application's total functionality that a user may decide to install independently.
- A component is a piece of the application or product to be installed.

The Windows Installer always installs or removes a component from a user's computer as a coherent piece. Components are usually hidden from the user. When a user selects a feature for installation, the installer determines which components must be installed to provide that feature.

Using Duplicate Package Identifiers

When you import a package into an Application Catalog database, Application Manager checks specific identifiers that are selected on the **Duplicate Package** tab of the ConflictSolver and Application Manager **Options** dialog box to determine if that package has already been imported (PackageCode, ProductCode, Product Language, ProductVersion, List of Transform Files, or—if no identifiers are selected—Product Name).

If Application Manager determines that you are attempting to import a duplicate package (based upon the selected identifiers), you are given the opportunity to enter a name of your choosing to identify the duplicate package before the package is imported. If the package is stored in the Software Repository, you can also choose to overwrite the existing version or treat it as a new version of an existing package.

Business Case for Importing a Package Multiple Times

You might encounter this situation if you are importing the same package into the same Application Catalog database multiple times, each time with a different set of transforms. One common reason why you might want to import the same package into your Application Catalog database more than once would be if you wanted to use Tuner to create custom installation SKUs of a common MSI package to distribute to different departments in your organization, each installation including certain features that are appropriate for the department and excluding certain features that are not appropriate. For example, if you were distributing a copy of Microsoft Office, you could add transforms to the Microsoft Office MSI package so that:

- Accounting's installation would include only Word and Excel
- Marketing's installation would include only Word and PowerPoint, and
- Development's installation would include only Word and Access.

Therefore, you might want to import the same package into your database more than once, each time with a different set of transformations. What happens when you import the package the second time depends upon the identifiers you selected on the **Duplicate Package** Tab. In this example:

- If you select the **List of Transform Files** and **ProductCode** identifiers on the **Duplicate Package** tab of the ConflictSolver and Application Manager **Options** dialog box, Application Manager will not identify these two packages as duplicate, even though they have the same **ProductCode**, because they have a different set of transformations. Therefore, the package will be imported with the same display name as the first package.
- If you only select the **ProductCode** identifier on the **Duplicate Package** tab of the ConflictSolver and Application Manager **Options** dialog box, Application Manager will identify the second package as a duplicate because the two packages have the same **ProductCode**.

Duplicate Product Name Conventions

When it identifies a duplicate package, by default Application Manager generates a new name for that package by pre-pending the Product Name with the Manufacturer's name and, if necessary, appending the Product Name with numbers:

- **1st Package:** PowerPoint
- **2nd Package:** Microsoft Corporation_PowerPoint

You can edit the **Duplicate Package Naming Syntax** on the **Duplicate Package** tab of the **Options** dialog box.

When Duplicate Packages are Identified During Bulk Import

If Application Manager is performing a bulk import or reimport, it still identifies duplicate packages using the user-specified criterion. Application Manager will generate a unique displayed product name and allow the import to proceed without prompting the user.

Application Manager identifies duplicate packages based upon the identifiers that you selected on the **Duplicate Package** tab of the ConflictSolver and Application Manager **Options** dialog box. See [Using Duplicate Package Identifiers](#) for more information.

Identifying Duplicate Package Identifiers

To identify duplicate package identifiers, perform the following steps.



Task:**To identify duplicate package identifiers:**

1. Select **Options** from the **Tools** menu. The ConflictSolver and Application Manager **Options** dialog box opens
2. Click the **Duplicate Package** tab.
3. Select the identifiers that you would like to use for product imports by selecting one or more of the following options:
 - **PackageCode Property**— Identifier of the package the product was installed from. No two non-identical .msi files should ever have the same package code.

- **ProductCode Property**—Unique identifier for the particular product release, represented as a string GUID, for example:
`{12345678-1234-1234-1234-123456789012}`
 - **Product Language**—The language the installer should use for any strings in the user interface that are not authored into the database.
 - **ProductVersion**—Version of the product in string format. The format of the string is: major.minor.build.
 - **List of Transform Files**—A list of the transformations associated with this package.
 - **[None Selected]**—If you do not select any of these five identifiers, Application Manager checks the Product Name identifier to determine if a package is a duplicate.
4. Click OK to save your changes and exit the **Options** dialog box
 5. Proceed with the package import, as described in [Importing Windows Installer Packages](#).



Note • If Application Manager is performing a bulk import or reimport, it still identifies duplicate packages using the user-specified criterion. Application Manager will generate a unique displayed product name and allow the import to proceed without prompting the user.



Note • The options that you select on the **Duplicate Package** tab of the ConflictSolver and Application Manager **Options** dialog box apply globally to all packages that you attempt to import; you cannot apply different identifiers to different packages. Also, since these options are saved in the AdminStudio Shared Directory, everyone using AdminStudio at your organization will share the same Duplicate Package options.

Performing an Ad-Hoc Import of Transform Files or Patch Files



Edition • You can only perform an ad-hoc import if you are connected to a Software Repository-enabled Application Catalog. The Software Repository feature is included in AdminStudio Enterprise Edition.

In a large enterprise, transform and patch files are created frequently to modify packages for different situations in their organization. For example, to localize a software application, a transform file is often created for each language supported by that application. Systems Administrators in these large enterprises have to keep track of which software package each file is supposed to be applied to.

You cannot tell which package a transform or patch file is associated with by just looking at the file. However, you can use the Import Wizard to perform an **ad-hoc import** of these loose transform and patch files into your Application Catalog. Then, Application Manager will:

- Determine which of the packages in the Application Catalog the transform or patch file is associated with, and
- Store the files in the Software Repository along with that package's other associated files.



Note • You can only perform an ad-hoc import if you are connected to a Software Repository-enabled Application Catalog. See [Using the Software Repository](#) for more information.



Note • To perform an ad-hoc import of a transform file or patch file, the package that the file is associated with must exist in the Application Catalog. If no associated package exists, the file cannot be imported.



Task: *To perform an ad-hoc import of a transform or patch file:*

1. Open **Application Manager**.
2. Connect to a Software Repository-enabled Application Catalog.
3. On the **Catalog** menu, click **Import Package**. The **Welcome** panel opens.
4. Click **Next**. The **File Selection** panel opens.
5. Select the transform or patch file that you want to import.
6. Select the **Add the file(s) to the Software Repository** option.

After you select this option, the **Additional Dependencies of the Import File** list appears, enabling you to select any dependencies of the file you are importing. For example, a dependency might be a file that the transform file references that is stored externally in a separate CAB file.

7. Select dependencies, if desired, and click **Next**.
8. The **Target Package Information** panel opens and lists the package that Application Manager has determined is associated with this file.



Note • For transforms and patches, if Application Manager determines that there are no packages associated with this file, you cannot proceed with the import.

9. Select the file's associated package.
10. If you want to reimport a new version of this package along with the file you are importing, select the **Reimport new version of the package with this file change** option. If you do not select this option, the package will not be reimported.
11. Click **Next**. The **Summary** panel opens, listing the **Adhoc file to import**, the **Destination Packages for the file**, and whether you chose to reimport the package.
12. Review the information in the **Summary** panel. If you are satisfied with the import options, click **Finish** to start the import. Progress messages appear in the **Output Window**.

About Transforms (.mst)

Application Manager supports the import of Windows Installer packages (.msi) with associated transforms (.mst). A transform is a collection of changes applied to an installation. By applying a transform to a base installation package, the installer can add or replace data in the installation database. The installer can only apply transforms during an installation.

The installer registers a list of transforms required by the product during the installation. The installer must apply these transforms to the product's installation package when configuring or installing the product.

A transform can modify information that is in any persistent table in the installer database. A transform can also add or remove persistent tables in the installer database. Transforms cannot modify any part of an installation package that is not in a database table, such as information in the summary information stream, information in substorages, information in nested installations, or files in embedded cabinets.

About Patches (.msp)

AdminStudio supports the import of Windows Installer packages (.msi) with associated patches (.msp). A Windows Installer patch (.msp file) is a file used to deliver updates to Windows Installer applications. A patch is a self-contained package that contains all the information required to update an application.

A patch package contains the actual updates to the application and describes which versions of the application can receive the patch. A patch package does not include a database like a regular installation package (.msi file). Patches contain at minimum two database transforms. One transform updates the information in the installation database of the application. The other transform adds information that the installer uses for patching files.

Importing Virtual Packages



Edition • Support for importing Microsoft App-V applications, ThinApp applications, and Citrix profiles into the Application Catalog is included in the AdminStudio Virtualization Pack.

You can import Microsoft App-V, VMware ThinApp, and Citrix virtual packages into the Application Catalog and associate them with their source Windows Installer package.

Virtual packages are self-contained entities which ordinarily cannot be modified after they are created. By associating a virtual package with the Windows Installer package which originated it, you have the convenience of being able to easily locate the virtual package's originating Windows Installer package, modify the original Windows Installer package, and then regenerate the virtual package.

You have several options in how to import a virtual package:

- **During the import of the Windows Installer package**—During the import of a Windows Installer package, if the Import Wizard searches for and finds a virtual package created by AdminStudio, you are prompted to also import that virtual package. See [Importing a Virtual Package During the Import of its Source Windows Installer Package](#).

- **After the import of the Windows Installer package**—If you import an AdminStudio-created virtual package into an Application Catalog that already contains the virtual package's source Windows Installer package, you are prompted to associate the virtual package with the Windows Installer package. See [Importing a Virtual Package After the Import of its Source Windows Installer Package](#).
- **By itself**—You can use the Import Wizard to import a virtual package directly into the Application Catalog. Then you can use the **Associate Package** function to associate the virtual package with its source package. See [Importing a Virtual Package Without Its Source Windows Installer Package](#) and [Manually Associating a Virtual Package with a Windows Installer Package](#).



Note • Application Manager does not perform a duplicate package check when importing virtual packages. Therefore, it is possible to have two or more copies of the same virtual package and/or name in the same Application Manager Group.

Importing a Virtual Package During the Import of its Source Windows Installer Package

When you are importing a Windows Installer package, if the Import Wizard finds a virtual package created by AdminStudio in a subdirectory of the directory containing that Windows Installer package, you are prompted to also import that virtual package.



Important • During virtual package conversion, AdminStudio 9.0 and later creates an XML file, *metadata.xml*, that contains metadata that identifies the original Windows Installer package that was used to create the virtual package. If a virtual package was created by a method other than the AdminStudio 9.x conversion process, this metadata file will not have been created, and therefore, the Import Wizard will not prompt you to import that virtual package even if it exists in a subdirectory of the directory containing a Windows Installer package that you are importing. Instead, you can perform the steps in [Importing a Virtual Package After the Import of its Source Windows Installer Package](#) and then [Manually Associating a Virtual Package with a Windows Installer Package](#).



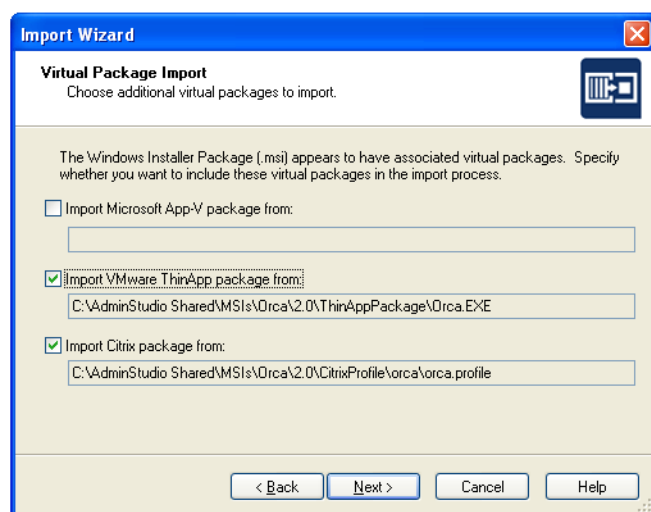
Task: **To import a virtual package during the import of a Windows Installer package:**

1. Open Application Manager.
2. Connect to your chosen Application Catalog.
3. On the **Catalog** menu, select **Import Package**. The Welcome panel of the **Import Wizard** opens.
4. Click **Next**. The **File Selection** panel opens.
5. Click **Browse** and select the Windows Installer package that has associated virtual packages that you want to import.



Important • The Import Wizard will only prompt you to import a Windows Installer package's associated virtual packages if those virtual packages are in a subdirectory of the directory containing the Windows Installer package.

6. Click **Next**. The **MST Source Information** panel opens.
7. Include any desired transform files, as described in [Importing Windows Installer Packages](#), and click **Next**. The **MSP Source Information** panel opens.
8. Include any desired patch files, as described in [Importing Windows Installer Packages](#), and click **Next**. The **Virtual Package Import** panel opens.



Note • This panel only opens if the Import Wizard finds a virtual package created by AdminStudio in a subdirectory of the directory containing the Windows Installer package you have selected to import.

If a Windows Installer package's virtual package is not in that location, you can use the Import Wizard to import the virtual package separately, as described in [Importing a Virtual Package After the Import of its Source Windows Installer Package](#), and then manually associate the two packages in Application Manager, as described in [Manually Associating a Virtual Package with a Windows Installer Package](#).

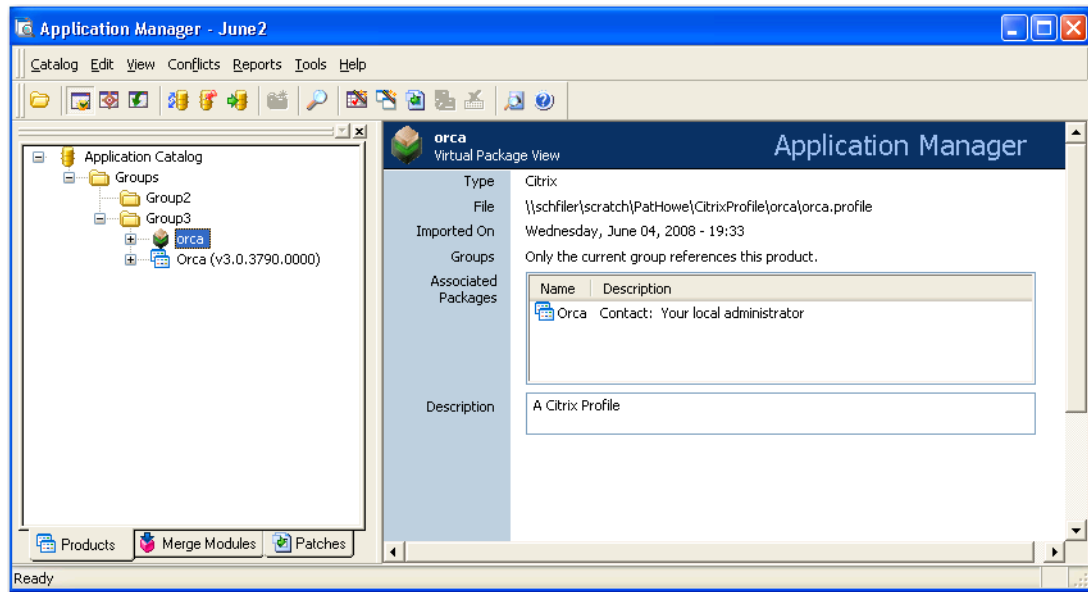
9. Select the virtual package(s) you want to import and click **Next**. The **Destination Group** panel opens.



Note • If you opened the Import Wizard by selecting a group folder and then selecting **Import** from the context menu, the **Destination Group** panel will not open.

10. Select the group or groups where you want to import this package, as described in [Importing Windows Installer Packages](#), and click **Next**. The **Summary** panel opens.

11. Click **Finish** to begin the import. The packages are imported. When import is complete, both the Windows Installer and the virtual package(s) are listed in the Application Catalog in the destination group you selected.
12. In Application Manager, select the virtual package that you have just imported. The **Virtual Package** view opens, displaying a list of all Windows Installer packages that are associated with that virtual package.



Note • Application Manager does not perform a duplicate package check when importing virtual packages. Therefore, it is possible to have two or more copies of the same virtual package and/or package name in the same Application Manager Group.

Importing a Virtual Package After the Import of its Source Windows Installer Package

During the import of an AdminStudio-created virtual package into an Application Catalog that already contains the virtual package's source Windows Installer package, the **Associate Virtual to MSI Packages** panel opens and you are prompted to associate the virtual package with the Windows Installer package.

If this source Windows Installer package has been imported into more than one location in the Application Catalog, multiple packages are listed.



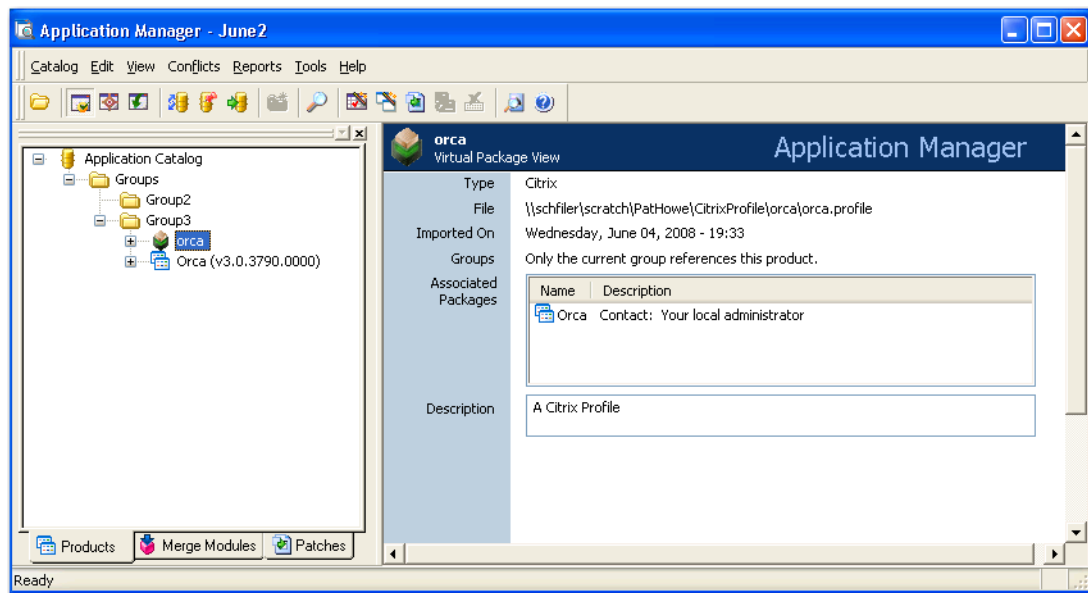
Important • During virtual package conversion, AdminStudio 9.0 and later creates an XML file, *metadata.xml*, that contains metadata that identifies the original Windows Installer package that was used to create the virtual package. If a virtual package was created by a method other than the AdminStudio 9.x conversion process, this metadata file will not have been created. Therefore, when importing this type of virtual package, the Import Wizard will not prompt

you to associate it with a Windows Installer package in the Application Catalog. Instead, you can perform the steps in [Manually Associating a Virtual Package with a Windows Installer Package](#).



Task: *To import a virtual package after the import of its source Windows Installer package:*

1. Open Application Manager.
2. Connect to your chosen Application Catalog.
3. On the **Catalog** menu, select **Import Package**. The Welcome panel of the **Import Wizard** opens.
4. Click **Next**. The **File Selection** panel opens.
5. Click **Browse** and select the virtual package that you want to import:
 - **Microsoft App-V application**—Select the App-V file with an .osd extension, such as MyApp.osd.
 - **VMware ThinApp application**—Select the ThinApp file with an .exe extension, such as MyApp.exe.
 - **Citrix profile**—Select the Citrix file with a .profile extension, such as MyApp.profile.
6. Click **Next**. The **Associate Virtual to MSI Packages** panel opens, prompting you to associate this virtual package with its source Windows Installer package.
7. Select the Windows Installer package and click **Next**. The **Summary** panel opens.
8. Click **Finish** to begin the import. The virtual package is imported. When import is complete, it is listed in the Application Catalog in the destination group you selected.
9. In Application Manager, select the virtual package that you have just imported. The **Virtual Package** view opens, and the associated Windows Installer package is listed in the **Associated Packages** list.



Importing a Virtual Package Without Its Source Windows Installer Package

You can import a virtual package into the Application Catalog even if you have not yet imported its source Windows Installer package. If you then import the virtual package's source Windows Installer package at a later date, you can use the **Associate Package** function in Application Manager to manually associate them.



Task: *To import a virtual package by itself:*

1. Open Application Manager.
2. Connect to your chosen Application Catalog.
3. On the **Catalog** menu, select **Import Package**. The Welcome panel of the **Import Wizard** opens.
4. Click **Next**. The **File Selection** panel opens.
5. Click **Browse** and select the virtual package that you want to import:
 - **Microsoft App-V application**—Select the App-V file with an .osd extension, such as MyApp.osd.
 - **VMware ThinApp application**—Select the ThinApp file with an .exe extension, such as MyApp.exe.
 - **Citrix profile**—Select the Citrix file with a .profile extension, such as MyApp.profile.
6. Click **Next**. The **Summary** panel opens.
7. Click **Finish** to begin the import. The virtual package is imported and it will not have an association to a Windows Installer package in the Application Catalog.

Manually Associating a Virtual Package with a Windows Installer Package

You can use the Import Wizard to import a virtual package directly into the Application Catalog even if you have not yet imported its source Windows Installer package. If you then import the virtual package's source Windows Installer package at a later date, you can use the **Associate Package** function in Application Manager to manually associate the virtual package with its source package.



Caution • After you have imported a virtual package into the Application Catalog, you are permitted to use the **Associate Package** function to associate it with **any** Windows Installer package in the Application Catalog, even one that is not its source package. Therefore, it is preferable to use the Import Wizard to import both the Windows Installer and virtual packages at the same time so that AdminStudio can create the proper associations.



Task: *To manually associate a virtual package with a Windows Installer package:*

1. Open Application Manager
2. Click on the virtual package you want to make an association for, and select **Associate Package** from the context menu. The **Virtual Package Association** dialog box opens, listing Windows Installer packages in the Application Catalog.
3. Select a Windows Installer package and click OK. The virtual package is now associated with the selected Windows Installer package.

Deleting a Virtual Package Association

You can delete a virtual package's association with a Windows Installer package by performing the following steps:



Task: *To delete a virtual package association:*


1. Open Application Manager.
2. Right-click on a virtual package, point to **Delete** and click **Package Association** on the context menu. The **Delete Virtual Package Association** dialog box opens, prompting you to confirm the deletion.
3. Select the association that you want to delete and click **OK**. The association is deleted.

Importing Merge Modules

For optimal performance, Merge Modules should be imported into an Application Catalog database prior to importing Windows Installer packages. This ensures that conflicts resulting from not using available merge modules are correctly identified.



Task: *To import Merge Modules:*

1. Open **Application Manager**.
2. Open the **Merge Modules** tab.
3. Select the root Merge Module or one of the imported Merge Modules and select **Import Merge Module** from the context menu. The **MSM Source Information** panel of the **Import Wizard** opens.
4. Click the Browse () button in the **Merge Modules** area and select the merge module file that you want to import.
5. To import multiple patches, you can repeat the procedure as necessary.
6. The order in which merge modules are applied can be changed by selecting a merge module in the list and clicking the Move Up and Move Down arrows.

7. If you need to delete a merge module you have added, clear its check box.
8. If you want to store this merge module in the Software Repository, select the **Add the file(s) to the Software Repository** option.
9. Click **Next**. The **Summary** panel opens.
10. Click **Finish** to accept these options and begin the import.

A report of the import process appears on the **Import** tab in the Output window.



Note • You can also import a Merge Module into the Application Catalog by selecting **Import Package** on the **Catalog** menu. However, using that method, only one Merge Module can be imported at a time.

About Merge Modules (.msm)

Application Manager supports the import of merge modules (.msm). Merge modules are essentially simplified Windows Installer .msi files.

A merge module cannot be installed alone because it lacks some vital database tables that are present in an installation database. Merge modules also contain additional tables that are unique to themselves. To install the information delivered by a merge module with an application, the module must first be merged into the application's .msi file.

Importing OS Snapshots

You can import OS Snapshot (.osc) files into the Application Catalog to use to determine conflicts between an operating system and a package. OS Snapshot files are files representing a particular computer system's contents. To generate an OS snapshot file, use the OS Snapshot Wizard to scan a computer's operating system and record the files, INI files, shortcuts, and registry entries present.



Caution • The OS Snapshot Wizard saves OS Snapshot information in two files: an .osc file (a collection of file type information) and an associated .nir file (registry information). The .nir file must be present in the same directory as the .osc file in order for import to be successful.



Task: **To import an OS Snapshot into an Application Catalog:**

1. Open **Application Manager**.
2. On the **Catalog** menu, click **Import Package**. The **Welcome** panel opens.
3. Click **Next**. The **File Selection** panel opens.
4. Select the OS Snapshot (.osc) file that you want to import

5. Click **Browse** and select the file that you want to import.
6. If you want to add this OS Snapshot to the Software Repository, so that it stores the OS Snapshot's associated .nir file, select **Add the file(s) to the Software Repository** option.
7. Click **Next**. The **Destination Group** panel opens.



Note • If you opened the Import Wizard by selecting a Group in the Application Catalog and then selecting **Import Package** from the context menu, the OS Snapshot will be imported into the selected group, and this panel will not appear.

8. Select one or more destination groups into which your OS snapshot will be imported. Click **Next**. The **Summary** panel opens.
9. Click **Finish** to accept these options and begin the import.

A report of the import process appears on the **Import** tab in the Output window.

About OS Snapshots (.osc)

Application Manager supports the import of OS Snapshot (.osc) files, which are files representing a particular computer system's contents. To generate an OS snapshot file, use the OS Snapshot Wizard to scan a computer's operating system and record the files, INI files, shortcuts, and registry entries present. The Wizard saves this information in an .osc file (a collection of file type information) and an associated .nir file (registry information).

When an OS Snapshot file is imported into Application Manager, it can be used as a reference point for conflict identification. See [Taking OS Snapshots](#) for more information.

To provide maximum flexibility during the OS Snapshot process, you can use the Exclusions Editor to create an exclusion list that identifies files, INI files, shortcuts, and registry entries that the OS Snapshot Wizard should disregard during the scan. Using this list, you can eliminate unnecessary files, shortcuts, or registry entries, and reduce the time it takes to perform the OS Snapshot. See [Configuring Exclusions Using the Exclusions Editor](#) for more information.



Caution • OS Snapshots should only be used for comparison in Application Manager. You should never attempt to convert an OS Snapshot into an MSI package.

Importing Marimba NCP Files



Edition • Import support for Marimba Native Channel Packager (.ncp) files is available in AdminStudio Enterprise Edition.

You can import Marimba NCP files into the Application Catalog. See [About Marimba Native Channel Packager Files \(.ncp\)](#) for more information.

To import Marimba NCP files into an Application Catalog, perform the following steps.



Task: *To import a Marimba NCP file into an Application Catalog:*

1. Open **Application Manager**.
2. On the **Catalog** menu, click **Import Package**. The **Welcome** panel opens.
3. Click **Next**. The **File Selection** panel opens.
4. Select a **Marimba NCP File (.ncp)** and click **Next**. The **Destination Group** panel opens.



Note • If you opened the Import Wizard by selecting a Group in the Application Catalog and then selecting **Import Package** from the context menu, the NCP File will be imported into the selected group, and this panel will not appear.

5. Select one or more destination groups into which your package will be imported and click **Next**. The **Summary** panel opens.
6. Review the information in the **Summary** panel . If you are satisfied with the import options, click **Finish** to start the import.

A report of the import process appears on the **Import** tab of the **Output** window. The Marimba NCP file package appears under the specified group in the **Application Manager Product View**.



Caution • The Marimba NCP file that should be imported into an Application Catalog is always named *manifest.ncp*. If you attempt to import another file with an .ncp extension into an Application Catalog, the import will fail.

About Marimba Native Channel Packager Files (.ncp)

Marimba NCP (Native Channel Packager) files are used to describe software components, their versions, their underlying structures, and their dependence on other components. Marimba's NCP file is derived from an OSD (Open Software Description) file and is currently used by Marimba's Castanet family of products.

Clients, such as Marimba Castanet Tuners, can use an NCP file to learn what files constitute a software package for particular software for a particular client platform and particular user preferences. Just as an HTML file can contain links to other HTML or media files, so can an OSD file contain links to other NCP files. These linked NCP files represent sub-packages or dependencies.

The following is an example of a very simple NCP file:

```
<SOFTPKG NAME=com.Installshield.www.AdminStudio"
  VERSION="1,0,0,0">
<TITLE>Solitaire</TITLE>
<ABSTRACT>AdminStudio by InstallShield Corporation</ABSTRACT>
<LICENSE HREF=http://www.installshield.com/AdminStudio
  license.html/>
<IMPLEMENTATION>
<OS VALUE="WinNT">
<OSVERSION VALUE="4,0,0,0"/></OS>
<OS VALUE="Win95">
<PROCESSOR VALUE="x86"/>
<LANGUAGE VALUE="en"/>
<CODEBASE HREF=http://www.installshield.com/AdminStudio.cab/>
</IMPLEMENTATION>
</SOFTPKG>
```

Application Manager supports the import of Marimba NCP Version 4.7.2 files into the AdminStudio Application Catalog, making them available for conflict identification and resolution. Please note the following:

- The Marimba NCP file that should be imported into Application Manager is *always* named manifest.ncp. If you attempt to import another file with an .NCP extension into Application Manager, the import will fail.
- Information from NCP files is stored in OS Snapshot specific tables within the Application Catalog.
- An NCP file can be selected as a source and a target package for Conflict detection, using [ACE23](#) and [ACE24](#). However, reporting, validation, and conflict resolution are not supported for the NCP package type.
- Extended attributes are supported for NCP packages.
- You can import NCP files from the command line.

Importing Other Setup Types

AdminStudio supports the import of wide variety of other (i.e., non-MSI) setup formats into the Application Catalog. See [About Other Setup Types](#) for a description of those setup types.

To import a setup in one of these formats into an Application Catalog, perform the following steps:



Caution • Package Auto Import, Application Catalog Replication, reporting, validation, conflict detection, and thus conflict persistence are not supported by the Other Setup Types package type. However, the Merge process does support this package type. See [Merging Application Catalogs Using the Merge Wizard](#) for more information.



Task: *To import Other Setup Types into an Application Catalog:*

1. Open **Application Manager**.
2. On the **Catalog** menu, click **Import Package**. The **Welcome** panel opens.
3. Click **Next**. The **File Selection** panel opens.
4. Select the non-MSI package that you want to import and click **Next**. The **Additional non-MSI Import Options** panel opens, where you can enter a name for the package and the base directory for the package (where all of its installation files are located).
5. Enter a **Product Name** to identify this package.
6. Click the **Explore** button and select the base **Directory** of this package where all of its associated files can be found.
7. If you want to import all the files in the specified directory folder and all of its subfolders, select the **Include entire directory hierarchy** check box. If you want to import only the files in the root of the specified directory folder, do not select this option.
8. Click **Next**. The **Destination Group** panel opens.



Note • If you opened the Import Wizard by selecting a Group in the Application Catalog and then selecting **Import Package** from the context menu, the file will be imported into the selected group, and this panel will not appear.

9. Select one or more destination groups into which this package will be imported and click **Next**. The **Summary** panel opens.
10. Review the information in the **Summary** panel. If you are satisfied with the import options, click **Finish** to start the import.

A report of the import process appears on the **Import** tab of the **Output** window. The Other Setup Type package appears under the specified group in the **Application Manager Product View**.

About Other Setup Types

A wide variety of other (i.e., non-MSI) setup formats exist. Application Manager supports the import of non-MSI setup types (such as InstallShield Professional or ISMP installations) into the Application Catalog. By importing these other setup formats, you allow AdminStudio to manage these setups in a manner consistent with other MSI based packages.

When importing one of these setup formats using the Import Wizard, you specify a directory or a directory hierarchy, and Application Manager will scan any setup.exe or i setup.exe files for pertinent setup information.



Note • Package Auto Import, Application Catalog Replication, reporting, validation, conflict detection, and thus conflict persistence are not supported by the Other Setup Types package type. However, the Merge process does support this package type. See [Merging Application Catalogs Using the Merge Wizard](#) for more information.

Using the Software Repository



Edition • The Software Repository feature is available in AdminStudio Enterprise Edition.

A Windows Installer package is made up of many files that are executed when the setup is run. However, only the .msi file is imported into the Application Catalog database. To safeguard these additional files against alteration or being misplaced, you can choose to manage these files using the Software Repository.

The Software Repository gives you a secure, transparent storage system for your AdminStudio data, especially Windows Installer packages used in the enterprise.

The AdminStudio tools are tightly integrated into the Software Repository, particularly Application Manager and InstallShield Editor. Certain concepts within AdminStudio itself require the Software Repository in order to operate:

- **Package version management and history**—You can store multiple versions of a package in the Software Repository. Instead of reimporting a package that has changed, you can check out a package and then check it back in either as a new package version or you can overwrite the existing version. When a package is checked out, it cannot be modified by another user. You can also view version history and include it in reports. See [Using Version Management Features](#) and [Viewing Package History](#) for more information.
- **Ad-hoc import of transform and patch files**—You can perform an ad-hoc import of transform and patch files into the Application Catalog and Application manager will determine which software package in the Application Catalog the transform or patch file is associated with. The file will then be stored in the Software Repository along with that package's other associated files. See [Performing an Ad-Hoc Import of Transform Files or Patch Files](#) for more information.

When a package is managed within the Software Repository, the Windows Installer file and all of its associated files and subfolders are imported into a subfolder of the Software Repository location identified for that Application Catalog.

Information on the AdminStudio Software Repository is presented in the following sections:

- [Enabling the Software Repository in Application Catalogs](#)
- [Methods to Import Packages into the Software Repository](#)
- [Identifying Software Repository Packages in Application Manager](#)
- [Using Version Management Features](#)
- [Software Repository Integration into Other AdminStudio Tools](#)

Enabling the Software Repository in Application Catalogs

To be able to import packages into the Software Repository of an Application Catalog, the Application Catalog must first have the Software Repository enabled. You can choose to enable the Software Repository feature when you are creating a new Application Catalog, and you can also enable it in existing Application Catalogs.

When an Application Catalog is enabled for the Software Repository, a Software Repository directory location must be specified.

When an Application Catalog has the Software Repository feature enabled, that does not mean that all of its existing packages and all newly imported packages will be automatically added to the Software Repository. Each time you import a package, you can choose whether you want to add it to the Software Repository.



Note • *If a package is not added to the Software Repository when it is imported, the only way to add it to the Software Repository after that is to import it again, and to choose to overwrite the existing package version.*

Enabling the Software Repository in a New Application Catalog

To enable the Software Repository in a new Application Catalog, perform the following steps.



Task: *To enable the Software Repository in a new Application Catalog:*

Create the new Application Catalog as described in [Creating New Application Catalogs](#), but on the [Select Software Repository Location Panel](#), do the following:

1. Choose the **Enable the Software Repository** option.
2. Select a **Software Repository Location**.
3. Enter a Login ID and Password of the **Proxy Account** that will be used to access this location.

The screenshot shows the 'Application Catalog Wizard' dialog box with the title bar 'Application Catalog Wizard'. The main heading is 'Select Software Repository Location' with a subtitle 'Select the location where the software repository should store imported packages.' Below this, there is a text block explaining the purpose of the software repository. A checkbox labeled 'Enable Software Repository' is checked. Below it, the 'Software Repository Location' is set to '\\server\softwarerepository\marketing'. A 'Proxy Account' section contains a 'Login ID' field with 'MyCompany\JohnSmith' and a 'Password' field with masked characters. At the bottom are buttons for '< Back', 'Next >', 'Cancel', and 'Help'.

Enabling the Software Repository in an Existing Application Catalog

To enable the Software Repository feature in an existing Application Catalog, perform the following steps.



Task: *To enable the Software Repository in an existing Application Catalog:*

1. Open the Application Catalog to which you want add Software Repository support in AdminStudio, ConflictSolver, or Application Manager.
2. On the **Catalog** menu, click **Properties**. The **Application Catalog Properties** dialog box opens.
3. Open the **Software Repository** tab.
4. Select the **Enable Software Repository** check box. The **Software Repository Location** option is enabled.

5. Browse to the location where you want to store the Software Repository for this Application Catalog.
6. Under **Proxy Account**, specify a **Login ID** and **Password** that will be used to access this location.
7. Click **OK**.

Methods to Import Packages into the Software Repository

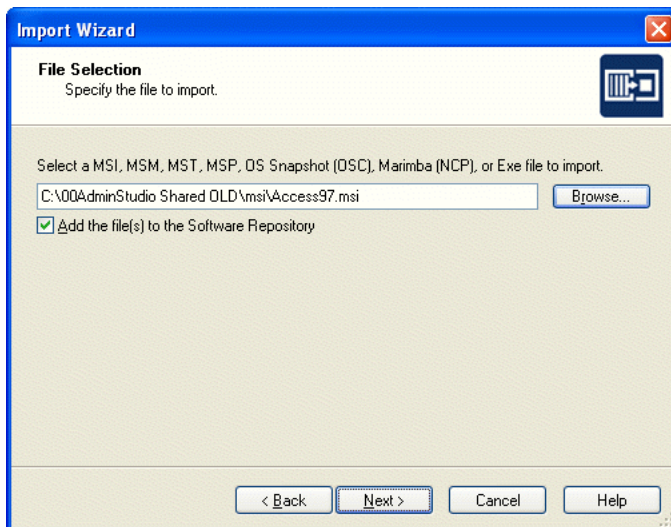
You can import packages into the Software Repository using the following methods:

- [Importing a Package into the Software Repository Using the Import Wizard](#)
- [Adding a Package to the Software Repository Using the ConflictSolver Process Assistant](#)
- [Adding a Package to the Software Repository by Overwriting Original Package](#)
- [Software Repository Auto Import Method](#)

Importing a Package into the Software Repository Using the Import Wizard

When you import a package into the Software Repository, all of its associated installation files are stored in the Software Repository location, not just the installation package itself.

To have the package and its associated files managed within the Software Repository, launch the Import Wizard as described in [Importing Data](#), and when the **File Selection** panel opens, select the **Add the file(s) to the Software Repository** option. If you are importing Merge Modules on the **Merge Modules** tab, this option is on the **MSM Source Information** panel.



When you click **Finish** on the Import Wizard, the selected package is imported into the Application Catalog, and all of that package's associated files and subfolders are copied into the Software Repository location specified for the Application Catalog. Progress messages are displayed in the Output Window stating that each file is being copied into the Software Repository.

Adding a Package to the Software Repository Using the ConflictSolver Process Assistant

When using the ConflictSolver Process Assistant to import a package, to add the package to the Software Repository, select the **Add package to the Software Repository** option on the **Organize Package** page.

In order to select this option, you must be connected to an Application Catalog that has the Software Repository enabled.

Adding a Package to the Software Repository by Overwriting Original Package

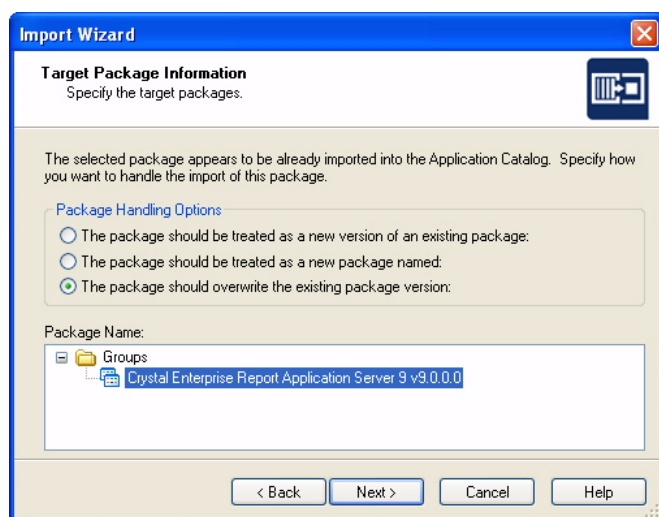
If a package is not added to the Software Repository when it is imported, the only way to add it to the Software Repository after that is to reimport it and overwrite the original package (assuming that the Application Catalog is Software Repository-enabled).



Task: *To add a package to the Software Repository by overwriting the original package:*

Reimport the package again by following the instructions in [Importing Windows Installer Packages](#), but this time:

1. Choose the **Add the file to the Software Repository** option on the **File Selection** panel
2. Choose the **The package should overwrite the existing package version** option on the **Target Package Information** panel:





Note • For more information on managing multiple versions of a package in the Software Repository, see [Using Version Management Features](#).

Software Repository Auto Import Method

Use the following command line options to perform a bulk import of Windows Installer packages into the Software Repository:

```
[General]  
ImportIntoTheSoftwareRepository=Y  
SuppressDuplicatePackageImport=N
```

Identifying Software Repository Packages in Application Manager

In Application Manager, packages that are managed within the Software Repository have a different icon than those that are not:

Table 6-6 • Application Manager Product View Icons Identifying Software Repository

Icon	Description
	Package is not managed within the Software Repository.
	Package is managed within the Software Repository.
	Package is managed within the Software Repository and is checked out.
	Merge Module is managed within the Software Repository.

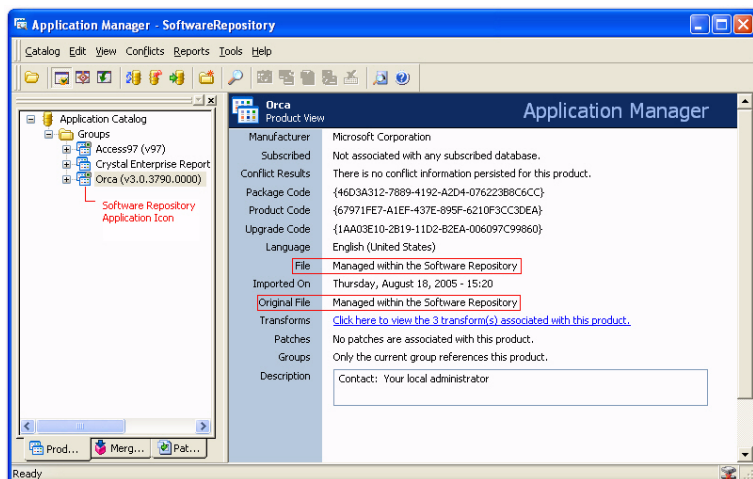


Note • These icons are not displayed in when viewing these packages in ConflictSolver.

Chapter 6: Managing Application Catalog Databases

Using the Software Repository

Also, when a Software Repository package is selected in the Application Manager **Product View**, the statement **Managed within the Software Repository** is listed for the **File** and **Original File** fields. Below is an example of a Software Repository package in the Application Manager Product View:



Using Version Management Features

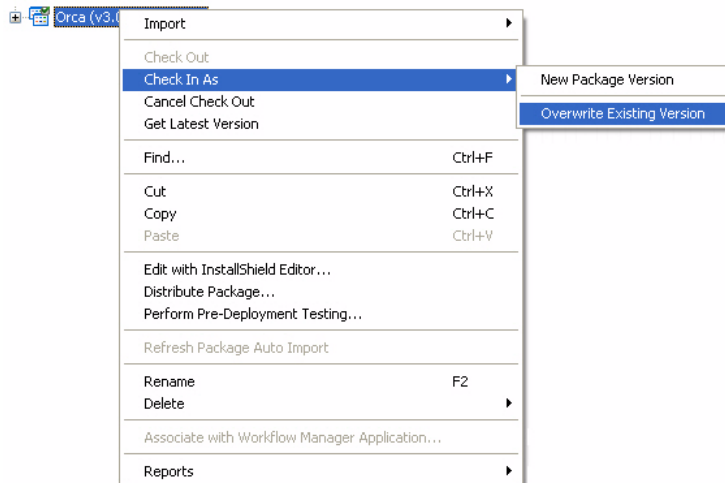
For those packages which are part of the Software Repository, you can store multiple versions of a package. Instead of reimporting a package that has changed, you can check out a package and then check it back in either as a New Package Version or you can Overwrite Existing Version.

The Software Repository version management features are described in the following topics:

- [Checking-Out and Checking-In Packages](#)
- [Cancel Check Out](#)
- [Getting a Copy of the Latest Version of a Package](#)

Checking-Out and Checking-In Packages

For those packages which are part of the Software Repository, you can store multiple versions of a package. Instead of reimporting a package that has changed, you can check out a package and then check it back in either as a New Package Version or you can Overwrite Existing Version.



When you check out a package, AdminStudio makes a local copy of the package files into the following directory:

C:\Documents and Settings\UserName\My Documents\AdminStudio\PackageName_PackageVersion

When a package is checked out, an icon is displayed on the Application Manager Product View to indicate that the package is checked out. If the current user is not the user who checked out the package, then the path portion of the property is not displayed.



When you check in a package, it will be imported into the Application Catalog and either the new package will be added as a new version of an existing package or the old package will be overwritten.

Cancel Check Out

If you checked out a package and then decide that you do not want to modify it, you can choose to cancel the check out by selecting the package, and then selecting **Cancel Check Out** from the context menu.

Unless a user is assigned to a Role with advanced Software Repository permissions, only the user who checked out the package will have the **Cancel Check Out** enabled. After the user confirms the operation, the local package files are deleted.

Getting a Copy of the Latest Version of a Package

If you want to get a copy of a package that is in the Software Repository, but you do not want to check it out, you can get a copy of that package by selecting the package, and then selecting **Get Latest Version** from the context menu. The package and all of its associated files will then be copied to the user's profile directory:

C:\Documents and Settings\UserName\My Documents\AdminStudio\PackageName

For example:

C:\Documents and Settings\JohnSmith\My Documents\AdminStudio\Adobe Reader



Note • The location where Software Repository files are copied to when you select **Get Latest Version** cannot be changed. The files are always copied to a subdirectory of the current user's My Documents directory.

Software Repository Integration into Other AdminStudio Tools

The Software Repository feature is integrated into several other AdminStudio tools:

- [InstallShield Editor](#)
- [ConflictSolver](#)
- [PackageExpert](#)
- [Distribution Wizard](#)
- [Predeployment Test Preparation Wizard](#)

InstallShield Editor

The Software Repository feature is integrated with InstallShield Editor in the following ways:

- You can launch InstallShield Editor from the Application Manager.
- From InstallShield Editor, you can browse for packages that are stored in the Software Repository and select them for edit. You can either check a file out for edit or simply get the latest version of the file to edit. You can also undo a check out from InstallShield Editor.
- From the InstallShield Editor, you can add a package to the Software Repository.
- You can add a package to the Software Repository via the InstallShield Editor build process.

For more information on InstallShield Editor's integration with AdminStudio, see [InstallShield Editor Integration with Application Manager and the Software Repository](#).

ConflictSolver

If you are performing conflict analysis on packages that are a part of the Software Repository, the resolution process will only operate against Checked Out packages by the current user. If a package is not checked out, you will be unable to automatically resolve any conflicts that are found or to modify the package in any way.

PackageExpert

If you are testing packages that are a part of the Software Repository, the resolution process will only operate against Checked Out packages by the current user. If a package is not checked out, you will be unable to automatically resolve any errors that are found or to modify the package in any way.

Distribution Wizard

You can launch the Distribution Wizard from Application Manager by selecting the package and then selecting **Distribute Package** from the context menu. When you do this, the package name in the **Windows Installer Package (.msi)** field of the **Installation Package Information** panel is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:

- **Not in the Software Repository**—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click **Browse** and select a different package.
- **In the Software Repository**—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed.

Predeployment Test Preparation Wizard

You can launch the Predeployment Test Preparation Wizard from Application Manager by selecting the package and then selecting **Perform Predeployment Testing** from the context menu. When you do this, the name in the **Source .MSI Package** field of the **Windows Installer File Selection** panel is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:

- **Not in the Software Repository**—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click **Browse** and select a different package.
- **In the Software Repository**—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed.

The **Target .MSI Package** field on the same panel is automatically populated with the name of the package specified in the **Source .MSI Package** field with `_analysis` appended to the end of the name. You can edit the name of this file, if desired. If the package is not in the Software Repository, the full name and path of the **Target .MSI Package** is displayed, but if the package is in the Software Repository, only the name of the **Target .MSI Package** is displayed (not the full path).

Sharing Application Catalog Data

The section describes the three methods for sharing Application Catalog data in your enterprise: merge, auto-import, and replication.

Table 6-7 • Sharing Application Catalog Data Help Topics

Section	Description
Merging Application Catalogs Using the Merge Wizard	You can use the Merge Wizard to merge data from a source Application Catalog into the currently open Application Catalog.
Automatically Importing Packages	You can use the Package Auto Import feature to link to MSI packages in a Remote Application Catalog, import the data into your local Application Catalog, and maintain consistency between the packages.
Replicating Application Catalogs	Using Application Catalog Replication simplifies software management by allowing you to selectively distribute subsets of data from a master Application Catalog to multiple IT groups at your organization.
Using Microsoft SQL Server to Perform Merge Replication of Application Catalogs	AdminStudio supports merge replication of Application Catalog databases using Microsoft SQL Server 2005. You can use merge replication to synchronize Application Catalog databases distributed in multiple locations across a large enterprise, ensuring that all databases are kept up-to-date.

AdminStudio provides three options for sharing Application Catalog Data:

- **Merging Application Catalog Information**—Use the **Merge Wizard** to merge the source Application Catalog's data into the currently open Application Catalog. The merged data is not linked in any way to the source Application Catalog.
- **Automatically Importing Packages**—The **Package Auto Import Wizard** offers two methods of automatically importing Windows Installer packages into your Application Catalog:
 - **Linking to Packages in a Remote Application Catalog**—Using the Remote Application Catalog option of the Package Auto Import Wizard, you can monitor Windows Installer packages in a Remote Application Catalog, and automatically import or re-import those packages at scheduled intervals. Application Manager can link to one or multiple Application Catalogs, and can link to one or multiple *packages* within those Application Catalogs.
 - **Automatically Importing Packages from a Network Directory**—Using the Network Directory option of the Package Auto Import Wizard, you can automatically import or re-import all Windows Installer packages in a specific folder into your Application Catalog at scheduled intervals.
- **Replicating Application Catalogs**—Use Application Catalog Replication's **Publication** and **Subscription** Wizards to distribute an Application Catalog database where it is needed across a large enterprise or multi-national corporation. Data is replicated between a Publisher Database Server and a Subscriber Database Server, which do not need to be available on the same network.

Comparison Chart

The following chart compares the three methods of sharing Application Catalog data:

Table 6-8 • Sharing Application Catalog Data Comparison Chart

Topic	Merging (Merge Wizard)	Auto Import Package Auto Import)	Replicating (Subscription and Publication)
AdminStudio Edition	Professional Edition	Professional Edition	Enterprise Edition
Configuration	Connection between two Application Catalog databases on the same network/domain.	Connection between two Application Catalogs Databases on the same network/domain or between an Application Catalog and a directory on the same network/domain.	Connection between: Publisher and Publisher Database Server, Publisher Database Server and Subscriber Database Server, and Subscriber and Subscriber Database Server
How It Works	The source Application Catalog's data is merged into the currently open Application Catalog. This is a one-time-only database merge. The merged data is <i>not</i> linked in any way to the source Application Catalog.	With the Remote Application Catalog option, you link to individual packages in a remote Application Catalog, and these packages are imported into your local Application Catalog. When the remote packages are modified, the packages in your local Application Catalog are automatically re-imported. When using the Network Directory option, if a modified version of a previously imported package is placed in the monitored Network Directory, the package is automatically re-imported.	An Application Catalog database is distributed across a large enterprise. Data is replicated between the Publisher Database Server and Subscriber Database Server, which do not need to be available on the same network.

Table 6-8 • Sharing Application Catalog Data Comparison Chart (cont.)

Topic	Merging (Merge Wizard)	Auto Import Package Auto Import)	Replicating (Subscription and Publication)
Customers Who Would Use This	Merging could be used in an organization where multiple people test packages. After the packages have been tested, all of the “tested” applications could be consolidated into one centralized Application Catalog.	Package Auto Import could be used to maintain consistency between applications in various Application Catalogs in organizations where they are satisfied with domain security and have both source and target Application Catalog databases (or source directory and target Application Catalog database) on an accessible domain.	Replication could be used at large enterprises that want to maintain a central Application Catalog, with smaller Application Catalogs at individual offices (such as New York, London, etc.). This method moves the data closer to the user (which improves performance), and allows the user to subscribe to only the data that they need (which makes it easier to work with).

Table 6-8 • Sharing Application Catalog Data Comparison Chart (cont.)

Topic	Merging (Merge Wizard)	Auto Import Package Auto Import)	Replicating (Subscription and Publication)
Advantages	<p>Very simple.</p> <p>Data is merged in with the data in the local Application Catalog.</p>	<p>Can select the packages you want to link to.</p> <p>Linked applications can be updated to match the source.</p> <p>Automatically imported data coexists with data in local Application Catalog.</p>	<p>Replication provides optimal security. Publisher Database Server and Subscriber Database Server do not need to be available on the same network. Operates within the SQL Server Security Model, making it a very secure way to replicate data—because the subscriber need not have access to the remote database.</p> <p>AdminStudio encapsulates functionality from SQL Server to support “Snapshot” type replication. Using AdminStudio to perform Snapshot replication is easier than trying to perform it using SQL Server Enterprise Manager.</p> <p>Can select individual packages in a Publication to subscribe to.</p> <p>Replication is very efficient. It operates within the SQL Server Job Refresh Model, which means that the replication/subscription process is run automatically.</p> <p>Automatically imported data coexists with data in the local Application Catalog.</p>

Table 6-8 • Sharing Application Catalog Data Comparison Chart (cont.)

Topic	Merging (Merge Wizard)	Auto Import Package Auto Import)	Replicating (Subscription and Publication)
Disadvantages	<p>Local user needs a physical connection to the database and has to have permissions on that remote database. This could allow the user performing the merge to modify the source database. This could cause a Security issue.</p> <p>This is a one time merge with no link to the source data and no updating.</p>	<p>When using the Remote Application Catalog option, the local user needs a physical connection to the database and has to have permissions on that remote database. This could allow the user performing the link to modify the source database. This could cause a Security issue.</p>	<p>You can subscribe to multiple subscriptions, but only one subscription can be active at any one time.</p>

Merging Application Catalogs Using the Merge Wizard

In some circumstances, you may need to merge data from two Application Catalogs. The reasons for this can vary, but usually involve increasing the number of packages available for conflict identification or the necessity of having the same data available in multiple geographic locations.



Task: *To merge data from two Application Catalogs:*

1. Open **Application Manager** and connect to the Application Catalog that is to contain the combined data.
2. On the **Catalog** menu, click **Merge** to open the **Merge Wizard**. The **Welcome** panel opens.
3. Click **Next**. The **Source Application Catalog** panel opens.
4. Select the Application Catalog that contains the data that you want to merge.

If the Application Catalog that you select requires additional authentication, the **Select Application Catalog** dialog box opens. Connect to that catalog as described in [Connecting to an Existing Application Catalog](#).

5. Click **Next** to advance to the **Groups** panel.
6. Select the groups you want to merge. You can also select to **Replace duplicate application data**.
7. Click **Next**. The **Progress** panel opens, where you can view the merge progress.
8. Click **Finish** to exit the Wizard.



Note • For a comparison of the three methods for sharing Application Catalog Data in AdminStudio, see [Sharing Application Catalog Data](#).

Automatically Importing Packages



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

For a large distributed enterprise, accurate and synchronized data is required for shared Windows Installer MSI packages. If a corporate level team maintains a database of packages used throughout the corporation in New York, then all the regional offices should be able to link to the corporate level MSI packages and get the different pieces of data.

Package Auto Import offers two methods of automatically importing Windows Installer packages into your Application Catalog and keeping them updated:

Table 6-9 • Package Auto Import Methods

Method	Description
Automatically Importing Packages in a Remote Application Catalog	Using the Remote Application Catalog option of the Package Auto Import Wizard , you can monitor Windows Installer packages in a Remote Application Catalog, and automatically import or re-import those packages at scheduled intervals. Application Manager can link to one or multiple Application Catalogs, and can link to one or multiple <i>packages</i> within those Application Catalogs.
Automatically Importing Packages from a Network Directory	Using the Network Directory option of the Package Auto Import Wizard , you can automatically import or re-import all Windows Installer packages in a specific Network Directory into your Application Catalog at scheduled intervals.



Caution • When automatically importing or re-importing packages from a Remote Application Catalog or a Network Directory using Package Auto Import, validation will not be performed even if the Check for Validation Errors option on the **Import** tab of the ConflictSolver or Application Manager Options dialog box is selected. If validation was performed during automatic import and if a package failed validation, it would not be imported into the Application Catalog, which would disrupt the automatic import process. Instead, you can perform validation after the package has been imported. See [Validating After Import](#).



Tip • When you link to a package in a Remote Application Catalog, Application Manager connects to the Remote Application Catalog and imports an MSI file into the local Application Catalog. In order for Package Auto Import to work properly, both the Remote Application Catalog and the Remote MSI file must be accessible to the linked users at all times. In other words, both the Remote Application Catalog and all of the MSI files that have been imported into it must be able to be accessed by linked users via a shared network drive.



Note • For a comparison of the different methods for sharing Application Catalog Data in AdminStudio, see [Sharing Application Catalog Data](#).

Automatically Importing Packages in a Remote Application Catalog



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

Using the Remote Application Catalog option of Package Auto Import, you can monitor Windows Installer packages in a Remote Application Catalog, and automatically import or re-import those packages at scheduled intervals.

The following topics are included in this section:

- [Linking to a Remote Application Catalog](#)
- [Editing a Link to a Remote Application Catalog](#)
- [Manually Updating a Linked Package](#)
- [Deleting a Link to a Remote Application Catalog](#)
- [Viewing Remote Package Links on the Application Manager Product View](#)

Linking to a Remote Application Catalog

Using the **Remote Application Catalog** option of **Package Auto Import**, you can monitor Windows Installer packages in a Remote Application Catalog, and automatically import or re-import those packages at scheduled intervals. Follow these procedures to configure a link to a Remote Application Catalog.



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.



Task: [To link to a Remote Application Catalog:](#)

1. Open **Application Manager**.
2. On the **Catalog** menu, select **Package Auto Import**. The **Package Auto Import dialog box** opens.
3. Click **Add**. The **Package Auto Import Wizard Welcome** panel opens.
4. Click **Next**. The **Package Auto Import Type Selection** panel opens, prompting you to specify whether you want to monitor packages in a **Remote Application Catalog** or monitor a **Network Directory**.

5. Select **Remote Application Catalog** and click **Next**. The **Remote Link Application Catalog** panel opens and lists Application Catalogs that you have recently accessed.
6. From the **Catalogs** list, select the Application Catalog you want to link to or click **Browse** to open the **Select Application Catalog** dialog box.

Connect to the selected catalog as described in [Connecting to an Existing Application Catalog](#).

7. Click **Next**. The **Groups** panel opens, listing all of the packages in the selected Application Catalog.
8. Select the packages within the Application Catalog that you want to link to.
9. Click **Next**. The **Schedule** panel opens.
10. To enable automatic import of the linked packages at scheduled intervals, select the **Schedule AdminStudio to update the changed packages** option. The scheduling fields are enabled.
11. From the **Run the process list**, select **Daily** or **Weekly**.
12. Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
13. From the **Starting on** list, click on the arrow to open the Calendar and then select the first day that you want automatic importing to begin.
14. Click **Next**. The **Summary** panel opens.
15. Review the summary of the options that were selected in the previous panels of the Wizard and click **Finish**. You are returned to the **Package Auto Import** dialog box and the new **Remote Application Catalog** link is now listed.



Note • Note that on the **Package Auto Import** dialog box, the **Start** button is disabled when this new Remote Application Catalog link is selected. This button is disabled until the linked packages have been imported into the local Application Catalog for the first time. The first automatic import of linked packages in a Remote Application Catalog is initiated when you click the **Close** button to exit the Package Auto Import dialog box.

16. Click **Close** to close the Package Auto Import dialog box.

Initial and Subsequent Automatic Imports

Import is initiated automatically when you set up the Remote Application Catalog connection, and then subsequent imports occur automatically at the scheduled date and time.

Initial Automatic Import

When you initially set up a connection to a Remote Application Catalog, automatic import is initiated when you click **Close** to exit the Package Auto Import dialog box. Import results are displayed in the **Output** window. The packages are imported into the **Package Auto Import Group** of the Application Catalog:

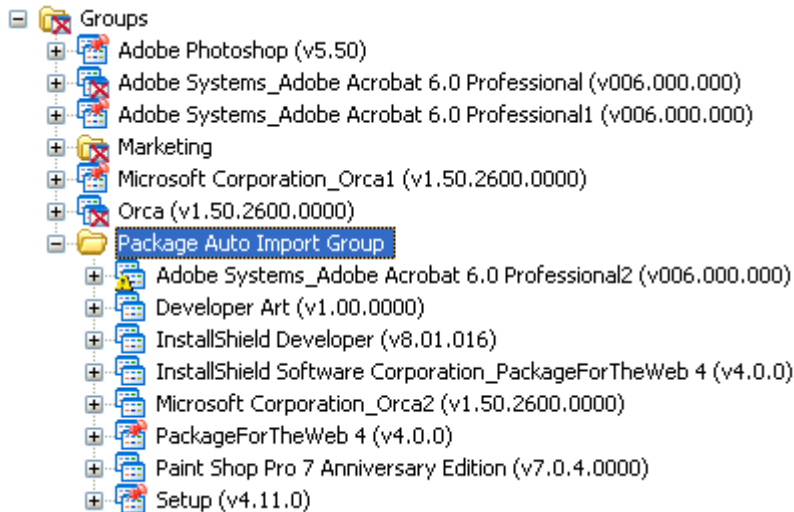


Figure 6-3: Package Auto Import Group of an Application Catalog

Subsequent Automatic Imports

After a package has been imported into the Application Catalog, an update check will be performed:

- At the scheduled interval (if the **Schedule AdminStudio to update the changed packages** option is selected), *and*
- Whenever you click **Start** on the **Package Auto Import dialog box**.

Preferred Authentication Modes

AdminStudio has preferred authentication modes when connecting to SQL Server Application Catalogs.

When you open AdminStudio, by default AdminStudio will attempt to connect to your SQL Server Application Catalog using NT authentication mode. If that fails, AdminStudio will display a login dialog box with the Server, Login ID, and Database fields pre-populated with your last selection.

If you use SQL Server Authentication Mode to login to your SQL Server Application Catalog, AdminStudio will not save SQL login information. Therefore, each time you want to refresh the data from this Application Catalog, you will have to supply AdminStudio with the login information. Therefore, it is recommended that you use NT authentication mode to login to SQL Server Application Catalogs.

Editing a Link to a Remote Application Catalog



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

On the Groups tab of the [Package Auto Import Properties \(Remote Application Catalog\) Dialog Box](#), you can link or unlink to different packages *within* an Application Catalog, and on the Options tab, you can modify update options and adjust the scheduled update interval.



Task: *To edit a Remote Application Catalog link:*

1. On the **Catalog** menu, select **Package Auto Import**. The **Package Auto Import** dialog box opens.
2. Select the **Remote Application Catalog** link that you would like to edit and click **Properties**.

The **Groups** tab of the **Package Auto Import Properties (Remote Application Catalog)** dialog box opens, displaying all of the groups and packages in the selected Remote Application Catalog that you can link to. Any package that you have already linked to is already selected.

3. Select or deselect check boxes to link or unlink to packages within this Remote Application Catalog.
4. If you want to specify options that directly affect the way the linked packages will be updated, select the **Options** tab.
5. On the **Options** tab, select one or more of the following **Refresh Options**:

- **Update package when its source is modified**
- **Delete package when the source package is deleted**

See [Manually Updating a Linked Package](#) for a detailed explanation of these two options.

6. To enable automatic import of the linked packages at scheduled intervals, select the **Schedule** option. The scheduling fields are enabled.

If you want to disable automatic importing of this package if it is modified in its Remote Application Catalog, leave the **Schedule** option unselected. With this option unselected, the only way that this package can be updated is to select the **Application Catalog** link on the **Package Auto Import** dialog box and then click **Start** to force an update.

7. From the **Run the process** list, select **Daily** or **Weekly**.
8. Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
9. From the **Starting on** list, click on the arrow to open the Calendar and then select the first day that you want automatic importing to begin.
10. Click **OK** to close the **Package Auto Import Properties** dialog box and then click **Close** to close the **Package Auto Import** dialog box.

Manually Updating a Linked Package



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

If you select the Remote Application Catalog link on the Package Auto Import dialog box and then click the Start button, the linked packages in the *local* Application Catalog are updated with the same packages in the *remote* Application Catalog. When you click on Start, how the packages are refreshed depends upon which options you have selected on the [Package Auto Import Properties \(Remote Application Catalog\) Dialog Box](#):

Table 6-10 • Import Options

Option	Description
Update package when its source is modified	<p>If you have selected this option:</p> <p>Application Manager will go and check if any of the linked packages have been modified in the Application Catalog.</p> <ul style="list-style-type: none">• If the linked package has been modified, then the local Application Catalog database is updated to reflect the changes.• If the linked package has not been modified, you receive a message stating that “No refresh is necessary”. <p>If you have not selected this option:</p> <p>Application Manager will still check to see if any of the linked packages have been modified:</p> <ul style="list-style-type: none">• If the linked package has been modified, Application Manager will inform you that the source package has been modified but that it will not update the local Application Catalog database because this option is not selected.• If the linked package has not been modified, you receive a message stating that “No refresh is necessary”.
Delete package when the source package is deleted	<p>If you have selected this option, Application Manager will delete any linked data in the local database if the data is deleted from the remote database.</p> <p>If you have not selected this option, even if the source package is deleted in the remote database, your copy of the package will not be deleted.</p>

As the linked packages are being re-imported, output messages will be displayed in the Package Auto Import tab of the Application Manager **Output** window.

Deleting a Link to a Remote Application Catalog



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.



Task: To delete a link to a Remote Application Catalog:

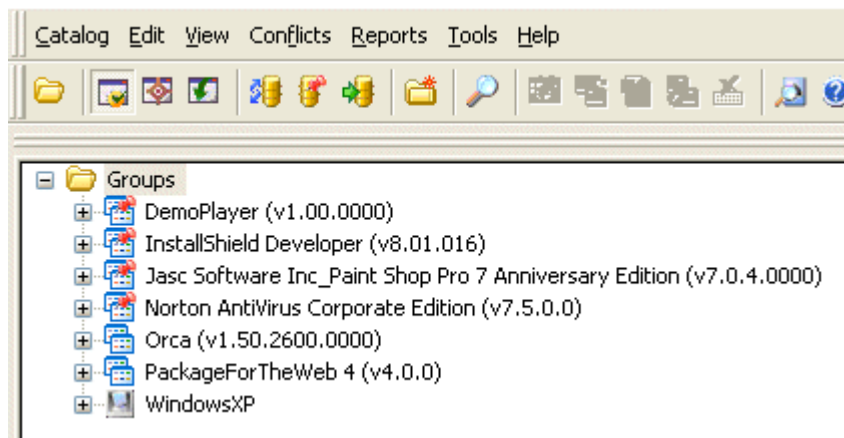
1. On the **Catalog** menu, select **Package Auto Import**. The **Package Auto Import** dialog box appears.
2. Select the **Remote Application Catalog** link that you would like to delete and click **Remove**.

Viewing Remote Package Links on the Application Manager Product View



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

The Application Manager Product View shows all the groups and packages present in an Application Catalog. The groups and packages that have been linked to using the Remote Application Catalog method of the [Package Auto Import Wizard](#) are displayed using a special icon, as shown below:



If a linked product in the local Application Catalog has been updated in the Remote Application Catalog and needs to be re-imported, an exclamation point is displayed on the icon (⚠).

Automatically Importing Packages from a Network Directory



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

Using the Network Directory option of Package Auto Import, you can monitor a directory location on the network (or a local directory) and automatically import or re-import packages in that directory at scheduled intervals.

The following topics are included in this section:

- [Linking to a Network Directory](#)
- [Editing a Link to a Network Directory](#)
- [Forcing an Import of Packages in a Network Directory](#)
- [Deleting a Link to a Network Directory](#)

Linking to a Network Directory

Using the **Network Directory** option of **Package Auto Import**, you can monitor a directory location on the network (or a local directory) and automatically import or re-import packages in that directory at scheduled intervals. Follow these procedures to configure a link to a network directory.



Task:

To link to a network directory:

1. On the **Catalog** menu, select **Package Auto Import**. The **Package Auto Import** dialog box opens.
2. Click **Add**. The **Package Auto Import Wizard Welcome** panel opens.
3. Click **Next**. The **Package Auto Import Type Selection** panel opens, prompting you to specify whether you want to monitor packages in a **Remote Application Catalog** or monitor a **Network Directory**.
4. Select **Network Directory** and click **Next**. The **Folder** panel opens.
5. Use the Browse button to select the network directory Folder that you want to monitor.



Note • While you are permitted to enter a local directory (rather than a network directory) as the monitored directory, this might cause problems if a person monitoring this directory has the same directory on his local machine. For example, if a directory named C:\MyAppCatalogs is chosen to be the monitored directory, and a person who choose to monitor this directory also has a directory on his local machine named C:\MyAppCatalogs, Application Manager would monitor that person's local directory rather than the directory that was chosen to be the monitored directory.

6. To also monitor the selected directory's sub-directories, select the **Also include sub-directories** option.

If you select this option, when packages in subdirectories are imported into your Application Catalog, they will be placed in groups that maintain the directory hierarchy. For example, if you are monitoring a directory named **Marketing**, and that directory has a subdirectory named **Graphics** that contains packages, when those packages are imported into your Application Catalog, they will be placed in the **Graphics** subgroup of the destination group.
7. Click **Next**. The **Destination Group** panel opens, prompting you to select the Application Catalog group into which the packages will be imported.
8. Select a destination group and click **Next**. The **Schedule** panel opens.
9. To enable automatic import of the linked packages at scheduled intervals, select the **Schedule AdminStudio to update the changed packages** option. The scheduling fields are enabled.

If you do not select this option, packages will only be imported when you select this entry on the Package Auto Import dialog box and then click **Start**.
10. From the **Run the process** list, select **Daily** or **Weekly**.
11. Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
12. From the **Starting on** list, click on the arrow to open the Calendar and then select the first day that you want automatic importing to begin.
13. Click **Next**. The **Summary** panel opens.
14. Review the summary of the options that were selected in the previous panels of the Wizard and click **Finish**. You are returned to the **Package Auto Import dialog box** and the new Network Directory link is now listed as a monitored activity.
15. To force an import of the packages in the monitored Network Directory, select the **Network Directory** entry and click **Start**. Import results will be displayed in the **Output** window. The packages will be imported into the specified Group in the Application Catalog:



Note • If you do not choose to perform a forced import, packages in the monitored Network Directory will be imported at the next scheduled interval.



Note • To modify Network Directory link settings, select the link on the [Package Auto Import Dialog Box](#) and click **Properties**. See [Editing a Link to a Remote Application Catalog](#).

Editing a Link to a Network Directory



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

On the **Groups** tab of the **Package Auto Import Properties (Network Directory)** dialog box, you can choose to monitor subdirectories and can adjust the scheduled update interval.



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.



Task:

To edit a link to a Network Directory:

1. On the **Catalog** menu, select **Package Auto Import**. The **Package Auto Import** dialog box opens.
2. Select the **Network Directory** link that you would like to edit and click Properties. The **Package Auto Import Properties (Network Directory)** dialog box opens.
3. To choose to monitor the selected folder's subfolders, select the **Also include sub-directories** option.

If you select this option, when packages in subdirectories are imported into your Application Catalog, they will be placed in groups that maintain the directory hierarchy. For example, if you are monitoring a directory named Marketing, and that directory has a subdirectory named Graphics that contains packages, when those packages are imported into your Application Catalog, they will be placed in the Graphics subgroup of the destination group.

4. Click **Next**. The **Destination Group** panel opens, prompting you to select the Application Catalog group into which the packages will be imported.
5. Select a destination group and click **Next**. The **Schedule** panel opens.
6. To enable automatic import of the packages in the Network Directory at scheduled intervals, select the **Schedule** option. The scheduling fields are enabled.

If you want to disable automatic importing of packages in the Network Directory, leave the **Schedule** option unselected. With this option unselected, the only way that packages in that directory will be imported into the Application Catalog is to select the **Network Directory** link on the **Package Auto Import** dialog box and then click **Start** to force an import.

7. From the **Run the process** list, select **Daily** or **Weekly**.
8. Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
9. From the **Starting on** list, click on the arrow to open the Calendar and then select the first day that you want automatic importing to begin.

10. Click **OK** to close the **Package Auto Import Properties** dialog box and then click **Close** to close the **Package Auto Import** dialog box.

Forcing an Import of Packages in a Network Directory



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

Select a monitored Network Directory on the Package Auto Import dialog box and then click the **Start** button to force an import of the packages in that Network Directory. As the packages are being imported (or re-imported if they have been modified), output messages are displayed in the Package Auto Import tab of the Application Manager Output Window.

Deleting a Link to a Network Directory



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.



Task: To delete a link to a remote package:

1. On the **Catalog** menu, click **Package Auto Import**. The **Package Auto Import** dialog box appears.
2. Select the monitored Network Directory that you would like to delete and click **Remove**.

The packages that have already been imported into your Application Catalog from this Network Directory will remain, but they will no longer be updated if a modified version of the package is placed in the Network Directory that they were imported from.

Package Auto Import and Duplicate Package Names



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

During manual importing of packages, if you are attempting to import a package that has been identified as a duplicate to a package already in the Application Catalog—based upon the identifiers you have selected on the **Duplicate Package** tab of the ConflictSolver and Application Manager [Options Dialog Box](#) (PackageCode, ProductCode, Product Language, ProductVersion or List of Transform Files), the [Target Package Information Panel](#) opens, proposing an automatically generated new name for the package and prompting you to accept the new name or enter one of your choosing.

However, if AdminStudio encounters a duplicate package name during auto import, the package you are importing will be automatically renamed by pre-pending the Product Name with the Manufacturer's name and, if necessary, appending the Product Name with numbers:

- **1st Package:** Application 2004
- **2nd Package:** ABC Corporation_Application 2004

The import is allowed to proceed without prompting the user.

You can edit the **Duplicate Package Naming Syntax** on the **Duplicate Package** tab of the **Options** dialog box.

Replicating Application Catalogs



Edition • Application Catalog replication is available in AdminStudio Enterprise Edition.

AdminStudio Application Catalog Replication makes it possible to distribute an AdminStudio Application Catalog database where it is needed across a large enterprise or multi-national corporation. One location can publish software packages to IT teams at other sites, keeping all teams current while enabling each to manage their software independently.

The following topics are included in this section:

Table 6-11 • Application Catalog Replication Topics

Topic	Description
About Application Catalog Replication	The benefits of using Replication, an overview of creating and managing Publisher and Subscriber Application Catalogs, and how to configure the AdminStudio Shared Location for Replication.
Publication/Subscription Lifecycle	Diagram illustrating how Application Catalogs are replicated.
Specialized User Scenarios	Describes how to subscribe to more than one Publication at a time, override the scheduled update day and time, and replicate Application Catalogs in controlled SQL database environments.
Creating and Managing Publications	Describes how to create, edit, publish, and delete a Publication. This section also explains how to set a Publication schedule and how to edit a Publication access list.
Creating and Managing Subscriptions	Describes how to create, enable/disable, manually update, and delete a Subscription.



Note • Only AdminStudio users who have been assigned sysadmin privileges to the currently connected SQL database server through the SQL Enterprise Manager can create Publications or set up Subscriptions using that database. For all other users, the Application Catalog Replication options are disabled.

About Application Catalog Replication

This section lists the benefits of using Replication, provides an overview of creating and managing Publisher and Subscriber Application Catalogs, and explains how to configure the AdminStudio Shared Location for Replication.

- [Benefits of Using Application Catalog Replication](#)
- [Creating and Managing Publisher and Subscriber Application Catalogs](#)
- [Configuring the AdminStudio Shared Location for Replication](#)
- [Initial Configuration Checklist](#)

Benefits of Using Application Catalog Replication

The benefits of using Application Catalog replication include the following:

- [Simplifies Software Package Management](#)
- [Allows You to Distribute Work Geographically Across the World](#)
- [Provides Optimal Security](#)
- [Automatically Performs SQL Server Snapshot Replication](#)

Simplifies Software Package Management

Using Application Catalog Replication simplifies software management by allowing you to selectively distribute subsets of data from a master Application Catalog to multiple IT groups at your organization.

One organization could have hundreds of applications, which would require multiple IT groups using AdminStudio to perform conflict detection, customization, repackaging, and testing. It would be unwieldy and confusing for all of those AdminStudio users to connect to the same Application Catalog.

In this situation, it would be very beneficial for one AdminStudio user to maintain a master Application Catalog in a central location, and then to replicate (distribute) subsets of the data to the other IT groups. By allowing an IT group to work with just the data that they need, performance and efficiency is improved.

Allows You to Distribute Work Geographically Across the World

For large multi-national enterprises, using Application Catalog Replication allows you to distribute work geographically across the world. A large enterprise could maintain a central Application Catalog that is updated as applications are generated and verified by multiple remote IT groups on local Application Catalogs. The verified applications could then be distributed globally from the master Application Catalog to all the distributed work groups of the enterprise.

Provides Optimal Security

Because the Publisher Database Server and Subscriber Database Server do not need to be available on the same network, Application Catalog Replication provides optimal security. Replication operates within the SQL Server Security Model, making it a very secure way to replicate data—because the Subscriber need not have any access to the remote database. To configure the Replication feature, the user must be assigned (using SQL Enterprise Manager) `sysadmin` privileges to the database server.

Automatically Performs SQL Server Snapshot Replication

AdminStudio Application Catalog Replication uses SQL Server Snapshot replication method. Snapshot replication operates within the SQL Server Job Refresh model, which means that the replication/refresh process is run automatically. Using AdminStudio to perform Snapshot replication is much easier than performing it using SQL Server Enterprise Manager.



Note • For more information on Snapshot Replication, see the [Snapshot Replication](#) topic on the Microsoft MSDN Web site.

Creating and Managing Publisher and Subscriber Application Catalogs

When you create an Application Catalog, it is designated as either a Publisher or a Subscriber catalog, and Application Manager is used to manage those subscriptions and publications.

- [Creating Publisher and Subscriber Application Catalogs](#)
- [Managing Publisher and Subscriber Application Catalogs](#)

Creating Publisher and Subscriber Application Catalogs

When creating a new SQL Server Application Catalog, you must choose (by selecting a check box) whether to define the Application Catalog as a Subscriber.

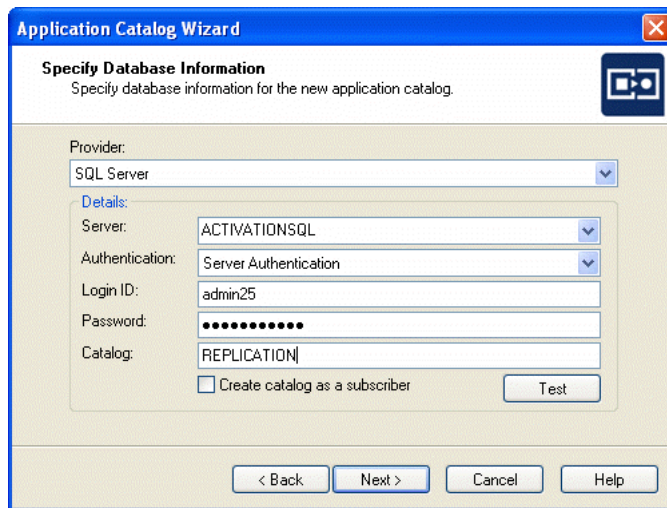


Figure 6-4: Create Database as a Subscriber Option

When this option is not selected, or when an existing database is upgraded, the Application Catalog is defined as a Publisher. The AdminStudio interface selectively enables/disables the Replication menu items based upon the type of Application Catalog that is currently opened.



Note • You can review your distributor properties in SQL Server Enterprise Manager on the Replication tab of the SQL Server Properties dialog box.

Managing Publisher and Subscriber Application Catalogs

In AdminStudio Application Catalog Replication, data is replicated between Publisher and Subscriber databases. You manage the Publisher and Subscriber databases using the following two dialog boxes:

- **Publication Manager dialog box**—Use to create and manage publications created on an AdminStudio Application Catalog database that is configured as a Publisher. See [Managing Publications](#).
- **Subscription Manager dialog box**—Use to create and manage Subscriptions to Publications on an AdminStudio Application Catalog configured as a Subscriber. See [Managing Subscriptions](#).

To see a diagram illustrating how data is replicated from the Publisher database to the Subscriber database, see [Publication/Subscription Lifecycle](#).



Note • The **Publication Manager** and **Subscription Manager** options under **Replication** on the AdminStudio **Catalog** menu are only enabled when a user with sysadmin privileges on an SQL Server database server connects to a database on that server.

- If that database was defined as a **Publisher** when it was created, the **Publication Manager** option is enabled.
- If that database was defined as a **Subscriber** when it was created, the **Subscription Manager** option is enabled.

See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.



Caution • You can create as many Subscriptions or Publications as you want to, but because any previously replicated data is deleted whenever new Subscription data is received, the Subscriber should have only one Subscription active at any one time. If you want to have more than one Subscription set up at one time and switch between them, see [Using the “Disable” Feature to Subscribe to More Than One Publication](#) for more information.

Configuring the AdminStudio Shared Location for Replication

When Publishing data using Application Catalog Replication, the Publication Server needs access to two script files that are installed in the AdminStudio Shared Directory Location. Therefore, the AdminStudio Shared directory location should refer to a network folder, making it accessible to the Publication Server. See [Specifying the Shared AdminStudio Application Catalog](#).

Initial Configuration Checklist

Before you can use Application Catalog Replication, you need to perform the following steps to configure the Subscriber and Distributor databases at your organization. If the Subscriber and Distributor databases are not configured in this manner, Replication will fail.



Note • The Publisher server is often configured as its own Distributor server.



Task: **To configure the Subscriber and Distributor servers:**

1. Check the SQL Server Agent logins configured on the Subscriber and Distributor databases, and make sure that the SQL Server Agent on the Subscriber database can login to the Distributor database via the Distributor SQL Server Agent using the distributor_admin login.
2. Configure a Distributor database for the servers that are publishing data.

Publication/Subscription Lifecycle

The Application Catalog database replication process is detailed in the diagram and notes below, including creating and publishing a Publication, and subscribing to that Publication.

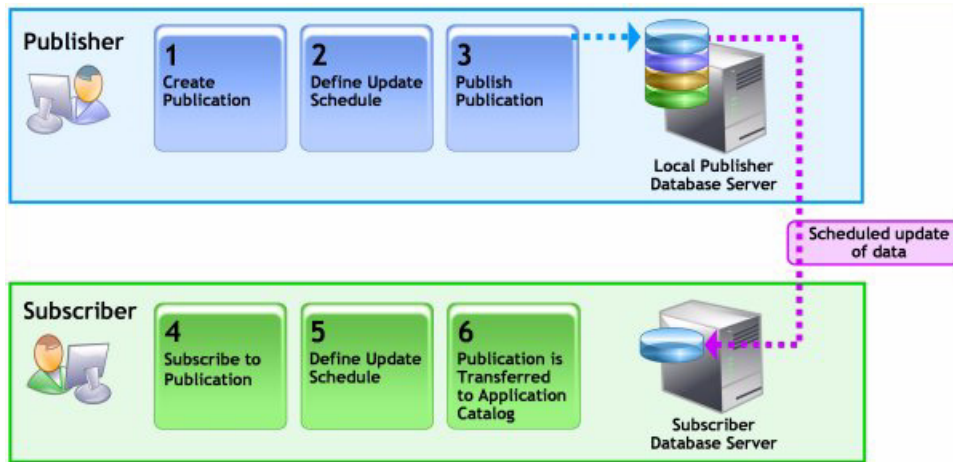


Figure 6-5: Application Catalog Database Replication Process

The following steps are included in the Application Catalog Database Replication Process:

Table 6-12 • Application Catalog Database Replication Process

Step	Description
Publisher Steps	
Step 1: Create Publication	The Publisher—an AdminStudio user who wants to make part of his Application Catalog available for Publication—creates a Publication on the local Publisher Database Server. The Publisher must have been assigned sysadmin privileges in SQL Enterprise Manager on the Publisher Database Server.
Step 2: Define Update Schedule	The Publisher defines the automatic update schedule for the Publication. At the specified date and time, the Publisher Database Server will check the Application Catalog to see if any published data has changed. If newer data exists, the Publication will be updated.
Step 3: Publish Publication	The Publisher generates the initial Publication by clicking the Run button on the Publication Manager dialog box. The Publication is now available for Subscription.

Table 6-12 • Application Catalog Database Replication Process (cont.)

Step	Description
Subscriber Steps	
Step 4: Subscribe to Publication	The Subscriber—an AdminStudio user who wants to receive data from the Publisher's Application Catalog—creates a Subscription to that Publication. The Subscriber must have been assigned sysadmin privileges in SQL Enterprise Manager on the Subscriber Database Server.
Step 5: Define Update Schedule	The Subscriber defines the automatic update schedule for the Subscription. At the specified date and time, the Subscriber Database Server will check to see if the Subscription on the Publisher Database Server has been updated. If it has been updated, the Subscription on the Subscriber Database Server will be updated.
Step 6: Data is Copied to Application Catalog	The subscribed Publication is copied to the Subscriber Database Server, and is listed under the Replication group in the Subscriber's Application Catalog (when viewed in the Application Manager Product View).

Terminology Review

The following terminology was used in this diagram and is used throughout the Application Catalog Replication help topics:

Table 6-13 • Application Catalog Replication Terminology

Term	Definition
Publisher	Person who creates the Publication.
Local Publisher Database Server	A Database Server making data available for replication.
Publication	The actual data made available for replication.
Subscriber	Person who subscribes to a publication.
Subscriber Database Server	A Database Server that receives the replicated data.
Subscription	A request to receive a Publication from a Distributor.

Specialized User Scenarios

The following topics explain how to use the Application Catalog Database Replication feature to best meet your organization's needs. The following topics are included:

- [Using the “Disable” Feature to Subscribe to More Than One Publication](#)
- [Overriding the Scheduled Update Day and Time](#)
- [Replicating Application Catalogs in Controlled SQL Database Environments](#)

Using the “Disable” Feature to Subscribe to More Than One Publication

At your organization, it could be beneficial to have more than one Subscription configured in the Subscription Manager, allowing you to switch from one Subscription to another, even though only one Subscription can be active and listed in your Application Catalog at any one time: the one that has been executed (updated) the most recently.

You might want to set up multiple Subscriptions in a situation where an IT group wants to work on a specific group of applications at a time. For example, an IT group might want to perform conflict analysis on Accounting applications one week and on Marketing applications the next week. When the IT group finishes work on the Accounting applications, the “Accounting” Subscription would be disabled and the “Marketing” Subscription would be enabled. By having multiple Subscriptions configured, local users can quickly switch between tasks.

On the **Subscription Manager** dialog box, Subscriptions that are enabled are preceded by a check mark (✓), while disabled Subscriptions are preceded by a red X (✗).



Note • You use the **Enable/Disable** button on the **Subscription Manager** dialog box to set the status of each Subscription to be either Enabled or Disabled. When you disable a Subscription, you are turning off its automatic update feature. Since Subscriptions are updated automatically, and since all of the currently active Subscription's data is replaced when a Subscription is updated, if you have multiple Subscriptions, it is to your advantage to disable all except the one you are currently using. This allows you to work on one set of data and control when that data is changed. If you have more than one Subscription enabled at a time, when the second enabled Subscription is automatically updated at its scheduled time, it will replace the data of the Subscription you were currently using.

Overriding the Scheduled Update Day and Time

If you want to manually update either Publication data or Subscription data before its scheduled update time, you can use the Run button on the Subscription Manager and Publication Manager dialog boxes. Using the **Run** button forces the selected Publication or Subscription to refresh itself.

Replicating Application Catalogs in Controlled SQL Database Environments

Some companies that have a controlled SQL database environment prefer to have their database administrators make major changes to their databases. In this case, you could select the Generate publication as script option on the **Publication Summary Panel** of the Publication Wizard or the Generate subscription as script option on the **Subscription Summary Panel** of the Subscription Wizard so that the Wizard would generate an SQL script of the code necessary to create this Publication or Subscription rather than actually create it.

When this option is selected, the script is the only output of the Wizard and the new item is not listed on the Publication Manager or Subscription Manager. The Database Administrator could then run this script to create the Publication or Subscription.

To create Publications and Subscriptions using SQL Server Enterprise Manager, perform the following steps:



Task: *To create publications and subscriptions using SQL Server Enterprise Manager:*

1. Use the **Publication Wizard** or **Subscription Wizard** to generate an SQL script by selecting the **Generate as script** option on the **Summary** panel.
2. Copy the script to your server.
3. Open **Query Analyzer**.
4. Select the database you want to work with.
5. Open the script file that was generated, and examine the `sp_addpublication` stored procedure to make sure that the `PreReplication.sql` and `PostReplication.sql` scripts are accessible from the system running Enterprise Manager. If they are not accessible, copy the scripts to an accessible location, and update the `sp_addpublication` stored procedure with the revised path names.
6. Execute the script.
7. Return to **SQL Server Enterprise Manager**.
8. If you created a Publication, open the database that you just generated the publication for and publish it.

Creating and Managing Publications

A Publication describes a subset of a database that you want to copy to a specific location to make it available to Subscribers. When a Publication is published, the data is copied to that accessible location. You create Publications to meet the needs of your subscribers. The Publication can be a partial Application Catalog—perhaps containing only 10 out of the 200 applications in your Application Catalog—or it can be an exact replica of your Application Catalog.

This section describes how to create, edit, publish, and delete a Publication, and also how to set a Publication schedule and edit a Publication access list. This section covers the following tasks:

- [Managing Publications](#)
- [Creating a New Publication](#)
- [Publishing a Publication](#)
- [Setting a Publication Schedule](#)
- [Editing a Publication](#)
- [Editing a Publication Access List](#)
- [Deleting a Publication](#)

Managing Publications

You create, publish, edit, and delete Publications using the **Publication Manager** dialog box.

The Publication Manager, which lists all Publications in the SQL Server database you are currently connected to, is accessed by pointing at **Replication** on the **Catalog** menu and selecting **Publication Manager**.



Note • The **Publication Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Publisher**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

The Publication Manager lists the last time that each of the Publications was updated (published to the Publisher Database Server) and the next scheduled update time for that Publication. From the Publication Manager, you can perform the following tasks:

- **Create a New Publication**—Click the New button to create a new Publication using the **Publication Wizard**.
- **Edit an Existing Publication**—Select a Publication and click Edit to access the **Publication Properties** dialog box, where you can edit the selected Publication's description, update schedule, user access list, and the packages included in the Publication.
- **Publish a Publication**—Select a Publication and click Run to manually update the selected Publication. The Publication data is replicated from your Application Catalog to the Publisher Database Server, making it available to Subscribers.
- **Delete a Publication**—Select a Publication and click Delete to delete the Publication data from the Publisher Database Server, making it no longer available for Subscription. When the list on the Publication Manager is refreshed, the Publication will no longer be listed.

Creating a New Publication

To create a Publication, you use the **Publication Wizard** to select the packages that you want to include in the Publication, specify the users that you want to be able to access the Publication, and schedule how often you want the Publication to be automatically updated. When the Publication Wizard is complete, the SQL Server is configured for this Publication, and the name of the Publication is listed on the **Publication Manager** dialog box.



Task: **To create a new Publication:**

1. On the Catalog menu, point to **Replication** and select **Publication Manager**. The **Publication Manager** dialog box appears.



Note • The **Publication Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Publisher**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

2. Click **New**. The **Welcome** panel of the Publication Wizard appears.
3. Click **Next**. The **Publication Details** panel appears.
4. In the **Name** text box, enter a name to identify this new Publication. This name will be listed on the Publication Manager dialog box.



Caution • The Publication Name cannot be a T-SQL reserved word.

5. In the **Description** text box, enter text to describe the purpose of this new Publication. This description will be listed on the **Publication Manager** dialog box.
6. Click **Change** next to the **Schedule** text box. The **Task Scheduling** dialog box appears.
7. In the **Run the process every** list, select how frequently you want this Publication to be automatically updated on the Publisher Database Server: **Day**, **Week**, or **Month**.
 - If you selected **Week**, select the day of the week that you want to update this Publication.
 - If you selected **Month**, select the day of the month that you want to update this Publication.
8. From the **Starting on** list, select the day or date that you want this Publication to begin being automatically updated on the day or date you selected from the Run the process every list.

For example, if you want the Publication to be automatically updated on the 15th of each month, you would select the first day of the first month that you want the automatic updates to begin, not the first day that the Publication should be updated. If it is Monday and you want the Publication to be updated every Friday, to have the Publication begin being updated on the upcoming Friday, you would select the current day of the week (Monday), but if you didn't want the Publication to be updated until the Friday after next, you would select Saturday.

9. Click OK to exit the Task Scheduling dialog box.
10. Click **Next**. The **Publication Data Options** panel appears.
11. Select the types of data you want to include in the Publication:
 - **Groups and Package Data**—Select this option if you want to select a specific set of packages from the various Groups in the Application Catalog to include in the Publication. If you select this option, the **Publication Packages** panel will appear, prompting you to select which packages you want to include in the Publication. If you do not select this option, all Groups and all packages within those Groups will be included in the Publication.
 - **Merge Module Data**—Select this option if you would also like to include all Merge Modules associated with the packages you have selected in the Publication. If you do not select this option, the Merge Module data will not be included in the Publication.
12. Click **Next**. The **Publication Access List** panel appears.
13. On this panel, specify the list of users allowed to access this Publication. It initially shows a list of users who have the necessary privileges to access the Publication: those users who have been assigned sysadmin privileges to this database server through the SQL Enterprise Manager. To remove a user, select it and click Remove.



Caution • Only users that have been assigned sysadmin privileges in SQL Enterprise Manager to the connected database server can establish the Publisher-Subscriber relationship between the databases and are therefore listed on this panel. You can remove a user from this list (so that they will not be able to access this Publication), but you cannot add additional users to this list using AdminStudio. To add new users, you must use SQL Enterprise Manager.

14. Click **Next**. If you selected the **Groups and Package Data** option, the **Publication Packages** panel appears.
15. On the **Publication Packages** panel, the Groups and Packages in the Publisher Application Catalog database you are connected to are listed. Select the Groups and packages that you want to include in this new Publication and click **Next**.

The **Publication Summary** panel appears, providing a detailed summary of the options that were selected in previous panels of the Wizard.
16. If you would like to generate an SQL script of the code necessary to create this Publication rather than actually create it, select the Generate publication as script option, and enter a path name for the script file.



Note • You might want to use this option if your organization has a controlled SQL database environment that prefers to have their database administrators make major changes to their databases. See [Replicating Application Catalogs in Controlled SQL Database Environments](#) for more information.

17. To choose to publish this Publication immediately upon completion of the Publication Wizard (instead of publishing it manually or waiting until its scheduled update), select the **Publish the publication immediately after creation** option.

18. Click **Finish** to complete the creation of this Publication.

- If you did not select the **Generate publication as script** option, the new Publication is listed on the Publication Manager.
- If you selected the **Generate publication as script** option, you will receive a message that the script was generated properly. When you close that message, you will see that this Publication is not listed on the Publication Manager. Request that your Database Administrator run the SQL script that you just generated in order to create this Publication.

19. Proceed with steps under [Publishing a Publication](#).

Publishing a Publication

When you publish a Publication, you are initiating a manual replication of the Publication data from your Application Catalog to the Publisher Database Server, making it available for Subscription. After its initial Publication, the Publication is also automatically “re-published” (updated) according to the schedule specified when the Publication was created.



Task:

To publish a Publication:

1. On the **Publication Manager**, select the Publication that you want to publish.



Note • If you did not select the *Generate publication as script* option on the **Publication Summary** panel of the Publication Wizard, the Publication should be listed on the Publication Manager as soon as the Publication Wizard finishes. If the Publication you want to publish is not listed here, make sure that the Publication Wizard was completed and, if the **Generate publication as script** option was selected, check with your Database Administrator to see if the script has been run.

2. Click **Run**.

The Publication data is replicated from your Application Catalog to the Publisher Database Server, making it available for Subscription.

Setting a Publication Schedule

When you create a Publication, you specify how frequently you want this Publication to be automatically updated on the Publisher Database Server. If you would like to change this frequency, you can edit the Publication from the Publication Manager dialog box.



Task: *To set a Publication schedule:*

1. On the **Publication Manager**, select the Publication that you want to edit and click Edit. The **Publication Properties** dialog box appears.
2. Click the Details tab.
3. Click Change next to the Schedule text box. The **Task Scheduling** dialog box appears.
4. In the Run the process every list, select how frequently you want this Publication to be automatically updated on the Publisher Database Server: Day, Week, or Month.
 - If you selected Week, select the day of the week that you want to update this Publication.
 - If you selected Month, select the day of the month that you want to update this Publication.
5. From the Starting on list, select the day or date that you want this Publication to begin being automatically updated on the day or date you selected from the Run the process every list.

For example, if you want the Publication to be automatically updated on the 15th of each month, you would select the first day of the first month that you want the automatic updates to begin, not the first day that the Publication should be updated. If it is Monday and you want the Publication to be updated every Friday, to have the Publication begin being updated on the upcoming Friday, you would select the current day of the week (Monday), but if you didn't want the Publication to be updated until the Friday after next, you would select Saturday.

6. Click OK to exit the Task Scheduling dialog box.
7. Click OK to exit the Publication Properties dialog box.

Editing a Publication

You can edit an existing Publication from the Publication Manager dialog box.



Task: *To edit a Publication:*

1. On the **Publication Manager**, select the Publication that you want to edit and click Edit. The **Publication Properties** dialog box appears.
2. Click the Details tab.
3. In the Description text box, enter text to describe the purpose of this Publication. This description will be listed on the Publication Manager and Subscription Manager dialog boxes.

4. In the Schedule field, specify how often you would like this Publication to be automatically updated. Click the Change button to access the **Task Scheduling** dialog box, where you can edit how frequently this Publication will be automatically updated.
5. If the Groups and Package Data option was selected on the **Publication Data Options** panel when this Publication was created, you can click the Packages tab and select or unselect packages for inclusion in the Publication.
6. Click OK to exit the Publication Properties dialog box.

Editing a Publication Access List

When you create a Publication, you specify the list of users allowed to access this Publication. If you would like to alter this list, you can edit the Publication from the Publication Manager dialog box.



Task: To edit a Publication Access List:

1. On the **Publication Manager**, select the Publication that you want to edit and click Edit.
The **Publication Properties** dialog box appears.
2. Click the **Access List** tab. The **Access List** tab lists users who have the necessary privileges to access the Publication: those users who have been assigned sysadmin privileges to this database server through the SQL Enterprise Manager.
3. Select the users allowed to access this Publication. To remove a user, select it and click **Remove**.



Note • Only users that have been assigned sysadmin privileges in SQL Enterprise Manager to the connected database server can establish the Publisher-Subscriber relationship between the databases and are therefore listed on this panel. You can remove a user from this list (so that they will not be able to access this Publication), but you cannot add additional users to this list using AdminStudio. To add new users, you must use SQL Enterprise Manager.

4. Click OK to exit the Publication Properties dialog box.

Deleting a Publication

When you no longer want to have a Publication available for Subscription, you can delete it using the Delete button on the **Publication Manager** dialog box.



Task: To delete a Publication:

1. On the **Publication Manager** dialog box, select the Publication that you want to delete.
2. Click Delete. A dialog box opens asking you to confirm the deletion.

3. Click OK.

The Publication data is deleted from the Publisher Database Server, and when the list on the Publication Manager is refreshed, the Publication will no longer be listed.

Creating and Managing Subscriptions

The topics in this section explain how to create a Subscription, edit its update schedule, enable or disable it, and update it manually. Please note the following about Subscriptions:

- **Subscribed data can co-exist with local data**—A Subscriber's Application Catalog can contain its own packages and can also contain subscribed Publications containing packages. The subscribed Publication is listed under the Replication group in the Subscriber's Application Catalog (when viewed in the Application Manager Product View).
- **Subscribed data is effectively read-only**—You should consider a Subscription as read-only data. Each time Subscription data is updated, it is completely replaced, so all changes that were made on a local copy of the Subscription data would be lost.

This section covers the following tasks:

- [Managing Subscriptions](#)
- [Creating a New Subscription](#)
- [Enabling/Disabling a Subscription](#)
- [Manually Updating a Subscription](#)
- [Deleting a Subscription](#)

Managing Subscriptions

You use the **Subscription Manager** dialog box to view Subscription information, create new Subscriptions, disable a Subscription so that it is not automatically updated, manually update a Subscription, and delete a Subscription.

The Subscription Manager, which lists all Publications that are currently subscribed to, is accessed by pointing at Replication on the Catalog menu and selecting Subscription Manager.



Note • The **Subscription Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Subscriber**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

The Subscription Manager lists the last time that each of the Subscriptions was updated, the next scheduled update time for that Subscription, and whether the Subscription is enabled (active) or disabled. From the Subscription Manager, you can perform the following tasks:

- **Create a New Subscription**—Click the **New** button to create a new Subscription using the **Subscription Wizard**.
- **Enable or Disable a Subscription**—Click the Enable/Disable button to toggle the selected Subscription's status between Enabled and Disabled. When you disable a Subscription, you are turning off its automatic update feature.



Note • While you can subscribe to multiple Subscriptions, only one Subscription can be active and listed in your Application Catalog at any one time: the one that has been executed (updated) the most recently. For information on how to use the Disable feature to subscribe to more than one Application Catalog, see [Specialized User Scenarios](#).

- **Manually Update a Subscription**—Click **Run** to manually update the selected Subscription. The Subscription Manager will then replace the data of the Application Catalog's currently active Subscription with the most recent version of the selected Subscription's data.
- **Delete a Subscription**—Click **Delete** to delete the selected Subscription from the Subscription Manager. When the list on the Subscription Manager is refreshed, the Subscription will no longer be listed.

Creating a New Subscription

To create a Subscription, use the [Subscription Wizard](#) to select the Publisher database that you want to subscribe to, and schedule how often you want the Subscription to be automatically updated. When the Subscription Wizard is complete, the replicated Publication data is copied into your Application Catalog, and the name of the Subscription is listed in the **Subscription Manager** dialog box.



Task:

To create a new Subscription:

1. On the Catalog menu, point to **Replication** and select **Subscription Manager**. The [Subscription Manager Dialog Box](#) opens.



Note • The **Subscription Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Subscriber**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

2. Click **New**. The [Welcome Panel](#) of the Subscription Wizard opens.
3. Click **Next**. The [Subscription Details Panel](#) opens, listing Application Catalogs that you have recently accessed.



Note • If you are connected to a standalone Application Catalog and you are not logged on to AdminStudio Enterprise Server, the [Login Required Dialog Box](#) opens before the Subscription Details panel opens, prompting you to log in.

4. From the **Catalogs** list, select the Publisher Application Catalog you want to subscribe to, or click **Browse** to open the [Select Application Catalog Dialog Box](#) and connect to an existing Publisher Application Catalog as described in [Connecting to an Existing Application Catalog](#).
5. To schedule how often the data will be updated, click **Change** next to the **Schedule** box. The [Task Scheduling Dialog Box](#) appears.
6. In the **Run the process every** list, select how frequently you want this Subscription to be automatically updated: **Day**, **Week**, or **Month**.
 - If you selected **Week**, select the day of the week that you want to update this Publication.
 - If you selected **Month**, select the day of the month that you want to update this Publication.
7. From the **Starting on** list, select the day or date that you want this Publication to begin being automatically updated on the day or date you selected from the **Run the process every** list.

For example, if you want the Publication to be automatically updated on the 15th of each month, you would select the first day of the first month that you want the automatic updates to begin, not the first day that the Publication should be updated. If it is Monday and you want the Publication to be updated every Friday, to have the Publication begin being updated on the upcoming Friday, you would select the current day of the week (Monday), but if you didn't want the Publication to be updated until the Friday after next, you would select Saturday.

8. Click **OK** to exit the **Task Scheduling** dialog box. You are returned to the **Subscription Details** panel, and the schedule you specified is now listed in the **Schedule** field.
9. Click **Next**. The [Subscription Selection Panel](#) appears, which displays the Publications available from the selected Publisher database that satisfy the [Publication Access List Panel](#) requirements for this user. In other words, it lists the Publications that you have permission to subscribe to.
10. Select one Publication from this list to subscribe to and click **Next**. The [Subscription Summary Panel](#) appears, providing a detailed summary of the options that were selected in previous panels of the Wizard.
11. If you would like to generate an SQL script of the code necessary to create this Subscription rather than actually create it, select the **Generate subscription as script** option, and enter a path name for the script file.



Note • You might want to use this option if your organization has a controlled SQL database environment that prefers to have their database administrators make major changes to their databases. See [Replicating Application Catalogs in Controlled SQL Database Environments](#) for more information.

12. Click **Finish** to complete the creation of this Subscription.

Enabling/Disabling a Subscription

You can use the Enable/Disable button on the **Subscription Manager** dialog box to set the status of each Subscription to be either Enabled or Disabled. When you disable a Subscription, you are turning off its automatic update feature. Subscriptions that are enabled are preceded by a check mark, while disabled Subscriptions are preceded by a red X.



Note • For information on how you can use the Disable feature to subscribe to more than one Publication, see [Specialized User Scenarios](#).



Task: To enable or disable a Subscription:

1. On the Catalog menu, point to Replication and select Subscription Manager. The **Subscription Manager** dialog box appears, listing all of your Subscriptions. Each Subscription's status (Enabled or Disabled) is displayed in the Last Run Status column.



Note • The **Subscription Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Subscriber**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

2. To disable an enabled Subscription, select it and click Disable.

The Subscription's status in the Last Run Status column is changed to Disabled, meaning that the Subscription will still be listed in the Subscription Manager, but will not be automatically updated.

3. To enable a disabled Subscription, select it and click Enable.



Note • This button toggles between Enable and Disable, depending upon the Status of the selected Subscription.

Manually Updating a Subscription

If you want to update a subscribed Publication to match the Publication on the Publisher Database Server and you do not want to wait until the next scheduled automatic update is performed, you can use the Run button on the **Subscription Manager** dialog box to manually update a Publication.



Task: **To delete a Subscription:**

1. On the **Subscription Manager**, select the Publication that you want to update.
2. Click Run. The Publication is replicated from the Publisher Database Server, replacing the earlier version.



Note • You can only update a subscribed Publication that has already been published at least once. If the Publication has never been published, the update will fail.

Deleting a Subscription

When you no longer want to subscribe to a Publication, you can Delete it from the Subscription Manager dialog box.



Task: **To delete a Subscription:**

1. On the **Subscription Manager**, select the Subscription that you want to delete.
2. Click Delete. A dialog box opens asking you to confirm the deletion.
3. Click OK. The Subscription's name is deleted from the Subscription Manager List.

Using Microsoft SQL Server to Perform Merge Replication of Application Catalogs

AdminStudio supports merge replication of Application Catalog databases using Microsoft SQL Server 2005. You can use merge replication to synchronize Application Catalog databases distributed in multiple locations across a large enterprise, ensuring that all databases are kept up-to-date. One location can publish software packages to IT teams at other sites, keeping all teams current while enabling each to manage their software independently.

To set up merge replication, first you use the AdminStudio interface to create an Application Catalog to serve as a Publisher database. You then use SQL Server Management Studio to configure this database as a Publisher, by creating a Publication. Lastly, you use SQL Server Management Studio to create one or more Subscriptions (and Subscriber databases).

Perform the following steps to implement merge replication of AdminStudio Application Catalogs:

- [Step 1: Creating the Publisher Database](#)
- [Step 2: Generating the Publication Script for the Publisher Database](#)
- [Step 3: Creating the Publication \(Running the Script\)](#)
- [Step 4: Creating a Subscription and a Subscribing Database](#)
- [Step 5: Starting the Snapshot Agent](#)
- [Step 6: Verifying the Existence and Integrity of the Subscriber Database:](#)

Step 1: Creating the Publisher Database

To create the Publisher database, perform the following steps;



Task:

To create the Publisher database:

1. Open AdminStudio and create a new AdminStudio Application Catalog on an SQL Server 2005 database server, such as the AdminStudio Application Catalog database server).
2. Close AdminStudio, and verify in SQL Server Explorer/Management Studio the existence of the newly created database.

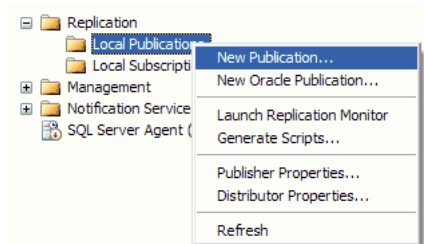
Step 2: Generating the Publication Script for the Publisher Database

To generate the Publication script for the Publisher database (Wizard), perform the following steps:



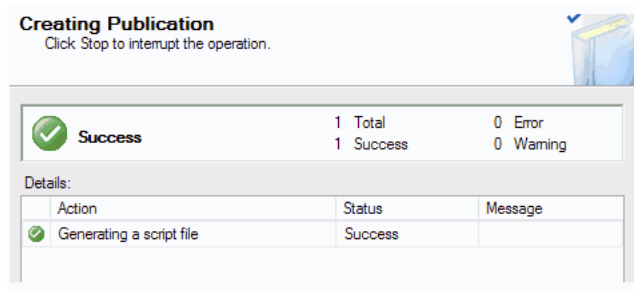
Task: *To generate the Publication script for the Publisher database:*

1. Locate the **Replication** node in Management Studio for the given database server.
2. Expand **Replication**, right-click on **Local Publications**, and select **New Publication Wizard** from the context menu:



3. When prompted to select the **Publication Database**, select the database you created in [Step 1: Creating the Publisher Database](#).
4. When prompted to select **Publication Type**, select **Merge Replication**.
5. When prompted to select **Subscriber Types**, verify that **SQL Server 2005** is the only option selected. (This document is targeted for SQL 2005 users only.)
6. On the **Articles** panel, select **Tables**, **Stored Procedures**, and **Views**, and click **Next**.
7. Continue by clicking **Next** on the **Article Issues** panel.
8. Continue by clicking **Filter Table Rows** on the **Article Issues** panel.
9. Configure the Snapshot agent as per your requirement. Leave defaults for default scheduling. (this setting can be changed later.)
10. On the **Agent Security** panel, click **Security Settings**, select **Run under the SQL Server Agent Service Account** option, and click **OK**.
11. Click **Next** on the Wizard.
12. On the **Wizard Actions** panel, clear the **Create the Publication** option and select the **Generate a script file** option.
13. Click **Next**.
14. Under **Script File Properties**, enter a valid path to store the script generated by this Wizard.
15. On the **Complete the Wizard** panel, enter a name for the Publication that you are creating (such as **ASPublication**), and click **Finish**.

16. Verify that the operation is reported as successful, as shown below:



Step 3: Creating the Publication (Running the Script)

To create the Publication, run the Publication script by performing the following steps:



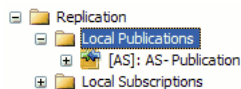
Task: *To create the Publication:*

1. Open the script created in [Step 2: Generating the Publication Script for the Publisher Database](#) in a text editor.
2. Replace this code:


```
@identityrangemanagementoption = N'auto', @pub_identity_range = 10000,
@identity_range = 1000, @threshold = 80
```

 with this code:


```
@identityrangemanagementoption = N'auto', @pub_identity_range = 5000000,
@identity_range = 5000000, @threshold = 80
```
3. Save the changes. There should be approximately 281 changes.
4. Run the script against the Publisher Database (created in [Step 1: Creating the Publisher Database](#))
5. Verify that the script runs without any errors or warnings.
6. Verify that a new publication is listed under **Replication/Local Publications**, as shown below. If no publication is listed, then refresh the **Replication** node.



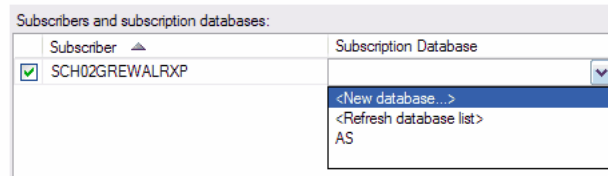
Step 4: Creating a Subscription and a Subscribing Database

To create a subscription and a subscribing database, perform the following steps:



Task: *To create a subscription and a subscribing database:*

1. Right-click on **Local Subscriptions** and select **New Subscription Wizard** from the context menu.
2. When prompted to select a **Publication**, choose the publication we created in [Step 3: Creating the Publication \(Running the Script\)](#).
3. Under **Merge Agent Location**, choose **Run all agents at the Distributor**. This setting can be the default, and can be changed by an Administrator to manage the server load.
4. When prompted to select **Subscribers and subscription databases**, select the server under **Subscriber** on the left side, and select **<New database...>** under **Subscription Database** on the right side, as shown below:



5. When prompted, enter a name for the new database (such as **AS-Subs 01**) and continue the wizard.
6. Under **Merge Agent Security**, provide the settings so that the Merge Agent runs under the default SQL Server Agent service account.
7. Under **Synchronization Schedule**, select **Run Continuously** under **Agent Schedule**. An administrator can modify this and provide a scheduling option instead of continuous.
8. Verify that **Initialize** is selected, and that **At first Synchronization** under **Initialize When** is selected. Click **Next**.
9. Click **Next**. Under **Wizard Actions**, select the **Create the subscription** option and then finish the Wizard.
10. Verify that there are no errors or warnings in the report.
11. Close the Wizard.
12. Refresh the **Publication** node under **Local Publications** and expand it. There should be a new **Subscription** listed under the **AS-Publication** publication.

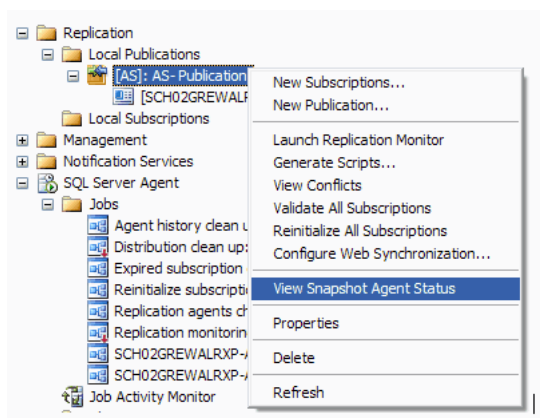
Step 5: Starting the Snapshot Agent

To start the Snapshot Agent, perform the following steps:

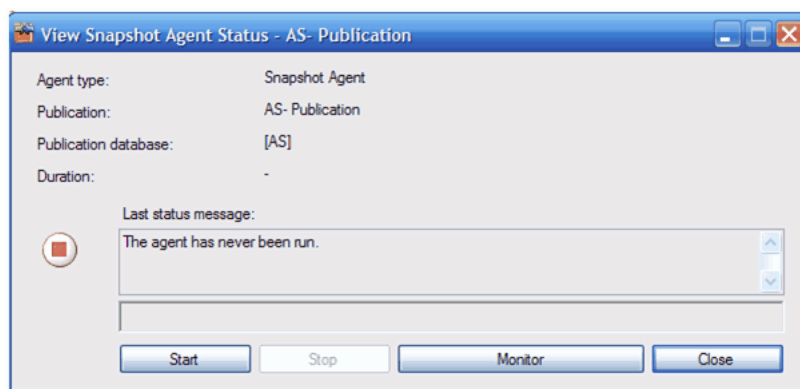


Task: *To start the Snapshot Agent:*

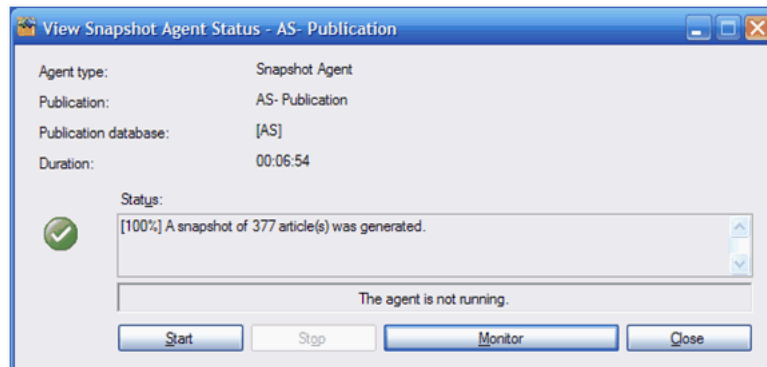
1. Right-click on **AS-Publication** and select **View Snapshot Agent Status** from the context menu.



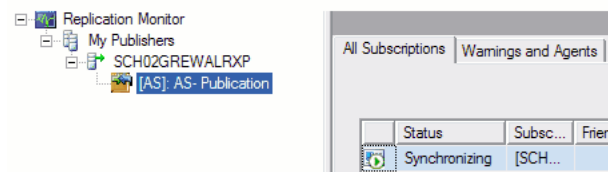
The **Snapshot Agent** opens:



- Wait until the Snapshot Agent is done taking the initial snapshot.



- Right-click on **Local Publications** and select **Launch Replication Monitor** from the context menu. Verify in the monitor that for the **AS-Publication** publication, there is one Subscriber and it is in **Synchronizing** state, as shown below:



Step 6: Verifying the Existence and Integrity of the Subscriber Database:

To verify the existence and integrity of the Subscriber database, perform the following steps:



Task: *To verify the existence and integrity of the Subscriber database:*

- Refresh and expand the **Database** node in Management Studio.
- Verify that a new database (created in [Step 1: Creating the Publisher Database](#)) is created. Expand the **Tables** under this database and verify that a list of AdminStudio tables appears.
- Launch AdminStudio and try to connect to this catalog.
- Launch Application Manager and try to import a package in this catalog.



Note • To create more subscribers, repeat [Step 4: Creating a Subscription and a Subscribing Database](#).

Taking OS Snapshots

The OS Snapshot Wizard provides a simple way to capture your basic operating system configuration. Using the OS Snapshot Wizard, you can scan your computer’s operating system and record the files, INI files, shortcuts, and registry entries present. The Wizard then creates an .osc file representing the system contents. You can import this snapshot file into the an Application Catalog database to identify potential conflicts between Windows Installer-based setups and your operating system.

To provide maximum flexibility during the OS Snapshot process, you can create an exclusion list (similar to the Repackager exclusion list) that identifies files, INI files, shortcuts, and registry entries that the OS Snapshot Wizard should disregard during the scan. Using this list, you can eliminate unnecessary files, shortcuts, or registry entries, and reduce the time it takes to perform the OS Snapshot.

The following topics are included in this section:

- [OS Snapshot Best Practices](#)
- [Configuring OS Snapshot Analysis Options](#)
- [Capturing an OS Snapshot](#)



Caution • OS Snapshots should only be used for comparison in ConflictSolver. You should never attempt to convert an OS Snapshot into an MSI package.

OS Snapshot Best Practices

Before capturing OS Snapshots, consider the following:

Table 6-14 • OS Snapshot Best Practices

Guideline	Description
Capture on a Clean System	For optimal OS Snapshots, you should only capture on a clean system. This ensures an accurate baseline of files necessary for the operating system. This also means that you should never attempt to capture other packages in addition to the operating system. Use Repackager when you need to capture applications.
Exit Other Applications	Shut down all other applications besides OS Snapshot prior to capture. Ideally, this should be done from the Windows Task Manager. This ensures that files are not locked during capture, and unnecessary temporary files are not inadvertently captured.
Only Use OS Snapshot for Import into Application Manager	Never attempt to convert an OS Snapshot file into a Windows Installer package to install an operating system. AdminStudio does not support this use of OS Snapshots.

Table 6-14 • OS Snapshot Best Practices (cont.)

Guideline	Description
OS Snapshots Take Time	Depending on the operating system configuration, OS Snapshot often takes a significant amount of time to capture the base OS state. Consider that many typical OS installations exceed 500MB and contain tens of thousands of files, translating into a lengthy operation of cataloging and converting the files into an OS Snapshot file.
Take Multiple OS Snapshots	If your environment contains either multiple operating systems, or variations on the same operating system, take snapshots of each OS or variation. You can store all of these in your Application Catalog, allowing you to make comparisons between MSI packages and each OS.
OS Snapshots and Repackaging Are Not the Same	OS Snapshots, as the name implies, is for snapshots of the operating system only. Repackaging is only for traditional installation packages. These operations, although similar, are still very specialized and should only be used for their respective purposes.

Configuring OS Snapshot Analysis Options



Task: *To configure analysis options for OS snapshot captures:*

1. Launch the **OS Snapshot Wizard** by clicking on its icon in the AdminStudio Tools Gallery. The **Welcome** panel opens.
2. Click **Next**. The **Project Information** panel opens.
3. Click **Edit**. The **Analysis Options** dialog box opens.
4. Configure the types of data you want to capture (Files, INI files, Shortcuts, and/or Registry data), and specify if you want to restrict the snapshot to a specific drive.
5. Click **OK** to return to the **Project Information** panel.
6. Click **Start**.

Capturing an OS Snapshot

You use the OS Snapshot Wizard to capture an OS Snapshots.

**Task:****To perform an OS Snapshot:**

1. Launch the **OS Snapshot Wizard** by clicking on its icon in the AdminStudio Tools Gallery. The **Welcome** panel opens.
2. Click **Next**. The **Project Information** panel opens.
3. Provide the OS Snapshot project name and OS Snapshot project folder for the OS Snapshot file.
4. Optionally, click **Edit** to configure analysis options. See [Configuring OS Snapshot Analysis Options](#).
5. Click **Start** to perform the OS Snapshot.
6. On completion of the OS Snapshot, review the results in the **Summary** panel.
7. Click **Finish**.



Note • The OS Snapshot is stored as an OSC file in the folder defined in the **Project Information** panel.

Reference

This section contains information on the all of the dialog boxes and wizards that are used when managing, connecting to, import data into, or sharing application catalogs.

- [Application Manager Views](#)
- [Dialog Boxes](#)
- [Wizards](#)
- [AdminStudio Database Schema Tables](#)
- [Application Manager Command-Line Functionality](#)

Application Manager Interface

The Application Manager user interface consists of three areas: the Navigation window, the Details pane, and the [Output Window](#). Both the Navigation window and the Output Window are dockable.

- **Navigation window**—The Navigation window consists of three tabs: **Products**, **Merge Modules**, and **Patches**.
- **Details pane**—When you select different items in these tabs, the Details pane displays corresponding information about that item.
- **Output window**—The Output Window consists of tabs where output is displayed during different Application Manager processes.

Application Manager also supports many context menus in addition to the menu bar and toolbar functionality.

This section details the Application Manager interface and includes the following topics:

- [Menus and Toolbar](#)
- [Product View Icons](#)
- [Merge Modules View Icons](#)
- [Context Menus](#)
- [Output Window](#)



Note • By default, Application Manager displays all conflict detection and resolution options. The display of the conflict detection and resolution options can be turned off by selecting the **Turn off display of conflict detection and resolution options in Application Manager** option on the **General** tab of the ConflictSolver and Application Manager **Options** dialog box. This section of the documentation describes Application Manager functionality. For information on ConflictSolver functionality, see [ConflictSolver Interface](#).

Menus and Toolbar

The following commands and toolbar buttons are available in Application Manager. Additionally, several commands can be accessed through [Context Menus](#).

Table 6-15 • Menus & Toolbars


Menu	Command	Shortcut	Button	Description
Catalog	Connect	Ctrl+O		Displays the Connect Application Catalog dialog box, where you can open an existing Application Catalog.

Table 6-15 • Menus & Toolbars (cont.)






Menu	Command	Shortcut	Button	Description
Catalog	Refresh	Ctrl + R		Refreshes the current Product and Merge Module views. This is particularly useful if multiple people are working on the same Application Catalog.
Catalog	Disconnect			Closes the currently open Application Catalog.
Catalog	Properties			Open the Application Catalog Properties dialog box.
Catalog	Change AES Password			Change the password of the current user to log in to the AdminStudio Enterprise Server.
Catalog	Logout			Log out of AdminStudio Enterprise Server.
Catalog	Package Auto Import	Ctrl + E		Link/unlink to one or multiple Application Catalogs. This allows you to copy packages between various Application Catalogs to maintain consistency between the copied packages. You can also choose to manually refresh the Application Catalog so that it remains consistent with other linked Application Catalogs.
Catalog	Merge	Alt + M		Launches the Merge Wizard, which you can use to select a different Application Catalog and import its contents into the current Application Catalog.
Catalog	Import Package	Ctrl + I		Launches the Import Wizard, allowing you to import Windows Installer packages, transforms, patches, merge modules, OS snapshots, Marimba NCP files, and virtual packages (Microsoft App-V, VMware ThinApp, and Citrix). You can also select Other Setup Types to import other non-MSI based setups.
Catalog	Exit			Closes Application Manager.
Edit	Find	Ctrl + F		Use to search for data in various tables in the Application Catalog.
View	Toolbar	Ctrl + 0		Toggles the Toolbar.

Table 6-15 • Menus & Toolbars (cont.)

Menu	Command	Shortcut	Button	Description
View	Status Bar	Ctrl+1		Toggles the Status Bar.
View	Output Window	Ctrl+2		Toggles the Output Window.
View	Application Manager	Ctrl+3		Select to switch to Application Manager.
View	ConflictSolver	Ctrl+4		Select to switch to the ConflictSolver.
View	Process Assistant	Ctrl+5		Select to switch to the ConflictSolver (Process Assistant).
Conflicts	Rules Viewer			Displays the Rules Viewer dialog box, from which you can create new ACE rules.
Reports	Package			Creates the Package Report.
Reports	Files			Creates the Files Report.
Reports	Registry			Creates the Registry Report.
Reports	Crystal Reports Conflicts			Creates the pre-defined Crystal Reports Conflicts Report.
Reports	Crystal Reports Files			Creates the pre-defined Crystal Reports Files Report.
Reports	Crystal Reports Registry			Creates the pre-defined Crystal Reports Registry Report.
Tools	Customize	Alt+O		Opens the Customize dialog box, where you can customize toolbar settings.
Tools	Options	Alt+N		Opens the ConflictSolver and Application Manager Options Dialog Box , where you can change various settings including default conflicts to identify, the default location for Windows Installer packages, report locations, the default validation file, and duplicate package identification options.
Help	Contents			Launches the online Help Library and displays the Contents tab.

Table 6-15 • Menus & Toolbars (cont.)

Menu	Command	Shortcut	Button	Description
Help	Index			Launches the online Help Library and displays the Index tab.
Help	Search			Launches the online Help Library and displays the Search tab.
Help	Support Central			Connects to the AdminStudio Support Web site.
Help	Web Community			Connects to the AdminStudio Web Community.
Help	ReadMe			Displays the AdminStudio ReadMe file.
Help	Feedback			Connects to an online form, through which you can provide feedback about AdminStudio.
Help	AdminStudio on the Web			Connects to the AdminStudio Web site.
Help	About Application Manager			Displays the About dialog box with version information.
	Help Library			Launches the online Help Library.

Product View Icons

These icons are used on the ConflictSolver and Application Manager Product Views:

Table 6-16 • ConflictSolver and Application Manager Product View Icons





Name	Icon	Description
Application Catalog		The Application Catalog that you are connected to. When you select it, the Application Catalog View opens and displays detailed summary information.
Group		A group, which is used to organize your data.
		A group containing a product with unresolved conflicts. See Resolving Conflicts .
Product		A product that has been imported into the Application Catalog database. The product could be an MSI file, an Other Setup Type package, or a Marimba NCP File.

Table 6-16 • ConflictSolver and Application Manager Product View Icons (cont.)











Name	Icon	Description
		A product with unresolved conflicts.
		A product that has an unresolved question with its associated files. One or more files, transforms, or patch files associated with a product is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Application Manager Product View identifying the files in question and prompting you to take action to resolve the problem.
		Package is managed within the Software Repository. This icon is only displayed in Application Manager.
		Package is managed within the Software Repository and is checked out. This icon is only displayed in Application Manager.
		A product with unresolved conflicts that also has an unresolved question with its associated files. One or more files, transforms, or patch files associated with a product is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Product View describing the missing files and prompting you to locate the missing files. When this icon is selected, a message appears in the Product View identifying the files in question and prompting you to take action to resolve the problem. Also, see Resolving Conflicts .
		A product linked to a Remote Application Catalog. See Automatically Importing Packages .
		A linked product in a local Application Catalog that has been updated on the Remote Server and needs to be refreshed in order to update its data with the data on the Remote Server. See Manually Updating a Linked Package .
		A virtual package (Microsoft App-V, VMware ThinApp, or Citrix) that has been imported into the Application Catalog.
		Virtual package is managed within the Software Repository. This icon is only displayed in Application Manager.
		Virtual package is managed within the Software Repository and is checked out. This icon is only displayed in Application Manager.

Table 6-16 • ConflictSolver and Application Manager Product View Icons (cont.)

















Name	Icon	Description
Transforms		A transform.
		A transform that has an unresolved question. It is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Application Manager Product View prompting you to take action to resolve the problem.
OS Snapshot		An OS snapshot, which is a file representing a particular computer system's contents.
Conflicts		Click to access the product's Conflicts View, where you can access information pertaining to the last execution of the Conflict Wizard.
Validation		Click to access the product's Validation View, which lists all ICE Errors and Warnings that were generated the last time the selected package was validated.
Patch Impacts		Click to access the product's Patch Impacts Analysis View, which lists patches for which there is patch impact data persisted against the product, and identifies the patch that caused the impact.
Dependencies		Click to display the Dependencies associated with the package. See Dependencies View for more information.
Extended Attributes		Click to display the optional Extended Attributes associated with the package. See Extended Attributes View for more information.
Files or Files/Components		Click to display the Files View or the Files and Components View, listing all of the files and components in the MSI package, merge module, OS snapshot, or NCP file.
		Click to display the Files View or the Files and Components View, listing all of the files and components in the MSI package, merge module, OS snapshot, or NCP file. The question mark icon indicates that a file is missing from the original import directory. When this icon is selected, a message appears in the Files View describing the missing files and prompting you to locate the missing files.
INI File Changes		Click to display the INI File Changes View , listing any INI file changes made by the product.
Merge Modules		Click to display any merge modules included the product.








Table 6-16 • ConflictSolver and Application Manager Product View Icons (cont.)

Name	Icon	Description
Associated Patches		Click to display the product's Associated Patches View , which displays a list of imported patches that, if installed, would update the selected product.
Registry		Click to display any registry entries created or changed by the product.
Shortcuts		Click to display any shortcuts created by the product.
Tables		Click to view the data for a given package in the Application Catalog.

Merge Modules View Icons

The following icons are used on the Merge Modules tab:

Table 6-17 • Merge Modules View Icons

Name	Icon	Description
Merge Modules		Indicates both the All Merge Modules view, the root node of the merge modules explorer containing a list of all merge modules in the Application Catalog, and an individual Merge Module.
Merge Module Managed Within the Software Repository		A Merge Module managed within in the Software Repository.
Components		Select to display any components created or changed by the merge module.
Dependency		Select to display any dependencies in the merge module.
Exclusion		Select to display any exclusions in the merge module.
Files		Select to display any files in the merge module
Products		Select to display any products in the Application Catalog that use the merge module.

Context Menus

Application Manager includes several context menus which can be accessed by right-clicking on nodes on the **Products**, **Merge Modules**, and **Patches** tabs. These menus provide specific functionality in relation to what is clicked.

- [Groups](#)
- [Packages](#)
- [Merge Modules Node](#)
- [Individual Merge Modules](#)
- [Patch View on the Patches Tab](#)

Groups

When you right-click on a group in the Application Manager Product View, the following commands are available through the context menu:

Table 6-18 • Group Context Menu Commands

Command	Description
Import	Allows you to import a package or an operating system patch into the Application Catalog. You have two options: <ul style="list-style-type: none">• Package—Select this option to launch the Import Wizard to import Windows Installer packages, transform files, merge modules or patch files (.msp) into the Application Catalog.• Patches—Select this option to launch OS Security Patch Wizard so that you can import Windows operating system and Microsoft application patches into the Application Catalog.
New Group	Creates a new group within the selected group.
Rename	Allows you to rename the selected group.
Delete	Removes the selected group from the Application Catalog, including any groups, products, or OS Snapshots contained within it.
Find	Select to perform a search for data in the various tables of the Application Catalog. See Searching an Application Catalog for more information.
Cut	Cuts the selected group from the Application Catalog. You can then paste it using the Paste command.
Paste	Pastes a cut group from Application Manager.
Properties	Opens the Group Properties dialog box.

Packages

When you right-click on a package in the Application Manager Product View, the following commands are available through the context menu:

Table 6-19 • Package Context Menu Commands




Command	Description
Import	<p>Allows you to import a package or an operating system patch into the Application Catalog. You have two options:</p> <ul style="list-style-type: none"> • Package—Select this option to launch the Import Wizard to import Windows Installer packages, transform files, merge modules or patch files (.msp) into the Application Catalog. • Patches—Select this option to launch OS Security Patch Wizard so that you can import Windows operating system and Microsoft application patches into the Application Catalog.
Reimport the package	<p>Reimports the package from its source location (if known and present).</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>
Find	<p>Select to perform a search for data in the various tables of the Application Catalog. See Searching an Application Catalog for more information.</p>
Cut	<p>Cuts the selected package from the Application Catalog. You can use the paste command to paste it to a new location.</p>
Copy	<p>Copies the selected package. You can use the paste command to paste it to a new location.</p>
Paste	<p>Allows you to paste a package that has been cut or copied.</p>
Edit with InstallShield Editor	<p>Open this package in InstallShield Editor, where you can directly edit the package or create a transform.</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>
Distribute Package	<p>Launch the Distribution Wizard.</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>

Table 6-19 • Package Context Menu Commands (cont.)






Command	Description
Perform Predeployment Testing	<p>Launch the Predeployment Test Preparation Wizard.</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>
Refresh Package Auto Import	<p>Rather than waiting until the next scheduled automatic update is performed, select this option to manually re-import a Windows Installer package that is being monitored in a Remote Application Catalog.</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>
Rename	Select to rename the selected package.
Delete	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Package—Removes the product from the current group. • Package from all Groups—Removes the product from all groups and removes it from the Application Catalog. • All Extended Attributes—Removes all extended attributes from the selected package. • Persisted Conflict Information—Removes persisted conflict information for the selected package. • Persisted Validation Information—Removes persisted validation information for the selected package. • History Log Information—Removes all change history data for the selected package.
Associate with Workflow Manager Application	<p>Launches the Associate with Workflow Manager Application dialog box, from which you can pick a package in Workflow Manager with which to associate the extended attribute data for the selected product.</p>  <hr/> <p>Note • Supported for Windows Installer packages only.</p>  <hr/> <p>Note • AdminStudio Workflow Manager is a Web-based application management system that has integrated functionality with AdminStudio.</p>

Table 6-19 • Package Context Menu Commands (cont.)

Command	Description
Reports	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Package—Creates the Package Report. • Files—Creates the Files Report. • Registry—Creates the Registry Report. • Crystal Reports Conflicts—Creates the pre-defined Crystal Reports Conflicts Report. • Crystal Reports Files—Creates the pre-defined Crystal Reports Files Report. • Crystal Reports Registry—Creates the pre-defined Crystal Reports Registry Report.  <p>Note • Supported for Windows Installer packages only.</p>

Merge Modules Node

When you right-click on the root-level Merge Modules node in the merge modules explorer, the following command is available through the context menu:

Table 6-20 • Merge Modules Node Context Menus

Command	Description
Import Merge Module	Launches the Import Wizard directly to the MSM Source Information panel.

Individual Merge Modules

When you click on an individual merge module in the merge modules explorer, the following commands are available through the context menu:

Table 6-21 • Individual Merge Modules Context Menus

Command	Description
Import Merge Module	Launches the Import Wizard directly to the MSM Source Information panel.
Delete	Removes the selected merge module from the Application Catalog.

Patch View on the Patches Tab

When you click on an individual patch in the [Patch View](#), the following commands are available through the context menu:

Table 6-22 • Patch View Context Menus

Command	Description
Rename	Rename the selected patch.
Delete	Removes the selected patch from the Application Catalog.
Cut	Cuts the selected patch from the Application Catalog. You can use the paste command to paste it to a new location.
Copy	Copies the selected patch. You can use the paste command to paste it to a new location.
Paste	Allows you to paste a patch that has been cut or copied.
Generate Report	Select to generate a Patch Report for that patch.
Properties	Launch the Patch Properties dialog box for that patch.

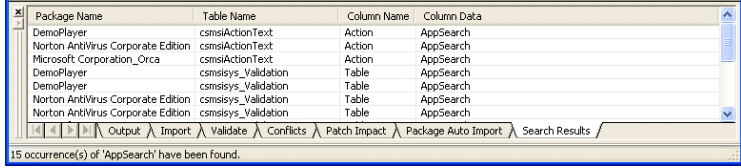
Output Window

When the Import and Package Auto Import Wizards are run, and when a Search is performed, the output messages and results of those wizards are displayed in the various tabs of the Output Window.

Table 6-23 • Output Window Context Menus

Tab	Description
Import Tab	When the Import Wizard is run to import a package into the Application Catalog, messages about the Import process are displayed in this tab.
Package Auto Import Tab	When the Package Auto Import Wizard is run to automatically import packages in a Remote Application Catalog or Network Directory, messages about the import process are displayed in this tab.

Table 6-23 • Output Window Context Menus (cont.)

Tab	Description
Search Results Tab	<p>When Find is used to search for data in Application Catalog tables, the data that is found is displayed in this tab, in the following format:</p>  <p>If you double click on this data, Application Manager will navigate to the appropriate record in the Tables View, and that record will be highlighted.</p>

User Permissions in Application Manager

Security and permissions can be assigned to Application Manager users to restrict the tasks that can be performed in Application Manager. The following table describes the various Application Manager permissions.

Table 6-24 • Application Manager Permissions

Permission	Explanation
Import Applications	Can import packages or merge modules into an Application Catalog, including re-importing packages resulting from unlinked data or conflict resolution
Delete	Can delete individual packages from the Application Catalog
Merge Database	Can merge information from two different Application Catalogs
Change Data	Can write to the Application Catalog. Without this permission, any Application Manager permission that allows users to write to the Application Catalog will not function; Application Manager is essentially read-only for roles without this permission.
Change Extended Attributes	Can change the XML file used to configure the extended attribute definition
Change Database	Can open another Application Catalog

Additionally, AdminStudio Administrators can restrict access to the Application Manager completely by disabling the Application Manager tool permission.



Note • Typically only AdminStudio Administrators and a few select users will have access to create new Application Catalogs, upgrade Application Catalogs, or otherwise affect multiple users.

Application Manager Views

The following views are associated with the Application Manager **Products** tab:

- [Application Catalog View](#)
- [Group View](#)
- [Product View](#)
- [OS Snapshot View](#)
- [Other Setup Types View](#)
- [Marimba NCP Files Views](#)
- [Patches Tab Views](#)
- [Merge Module Views](#)

Application Catalog View

The Application Catalog View is displayed on the right side of Application Manager whenever the Application Catalog node is selected. The following information is displayed:

Table 6-25 • Application Catalog View

Item	Description
Name	The name of the Application Catalog is displayed in the title bar.
Statistics	The number of packages, merge modules, OS Snapshots, patches, and groups is listed.
Most Recently Modified Packages	Packages that have been modified recently are listed with the most recently modified package listed first. If you double-click on a package, the Product View for that package is opened.
Application Catalog History	Operations that have been performed recently on the Application Catalog are listed, with the most recently performed listed first.

Group View

The Group view is displayed on the right side of Application Manager whenever a group is selected.

If you select a group or product in a group, properties for that group or product are displayed in the Group View.

Additionally, if you double-click an item in the Group View, the corresponding item in the Application Manager Product View is selected. You can sort the products by either Name or Description by clicking on the column title.

Product View


When you select a Windows Installer Package (.msi), details about that product are displayed in the right pane of the user interface.

The following information is included:

Table 6-26 • Application Manager Product View

Field	Description
Manufacturer	The manufacturer of the package.
Subscribed Database	If this package is associated with a subscribed database, the name of that database is listed here. If it is not, the following statement appears: Not associated with any subscribed database.
Conflict Results	If any persisted conflict data exists, this is a hyperlink to the package's Conflicts View.
Package Code	The globally unique identifier (GUID) for the setup package.
Product Code	A string that uniquely identifies the product.
Upgrade Code	A string used to upgrade the application. The upgrade code for a package groups that package into a specific product family.
Language	The language of this package.
File	Identifies the location of this package. It can be either a hard-coded path or a UNC path. If the package is part of the Software Repository, the following statement appears: Managed within the Software Repository

Table 6-26 • Application Manager Product View

Field	Description
Original File	<p>When a user selects a Windows Installer package (.msi) along with one or more patch files (.msp) to import, AdminStudio first performs an administrative installation to merge the .msi and .msp files into one .msi file, and then imports the merged .msi file into the Application Catalog. In this instance, the Original File field lists the name and path of the original Windows Installer package that the patches were applied to, while the File field lists the name and path of the merged .msi file that was imported.</p> <p>If the package is part of the Software Repository, the following statement appears:</p> <p>Managed within the Software Repository</p>  <p>Note • When a Windows Installer package that was imported without a patch is selected, the entry of the Original File and File fields is identical.</p>
Imported On	The date and time the package was imported.
Transforms	This can be either a hard-coded path or a UNC path.
Patches	The patch files that were used to get to the final imported MSI package.
Groups	Any groups to which the package belongs.
Description	You can edit this with additional information about the product.



Note • If the package and/or transforms are no longer in their original import directory, you can locate the file(s) from the provided hyperlink. You are also informed if the last modified date for the package in the Application Catalog does not match the last modified date of the package in its external location. You are given the opportunity to reimport the package to keep it synchronized in the Application Catalog.

If you click on the plus sign to expand a package in the Application Manager Product View, a node is listed for each available constituent view. For example, for a Windows Installer package, the following nodes are listed:

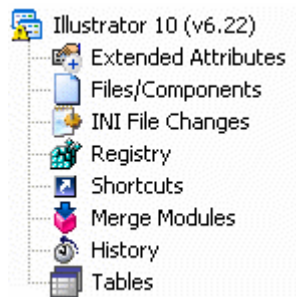


Figure 6-6: Application Manager Product View Nodes

When you select one of these nodes, a constituent view opens in the right pane:

- [Extended Attributes View](#)
- [Files/Components View](#)
- [INI File Changes View](#)
- [Registry View](#)
- [Shortcuts View](#)
- [Merge Modules View](#)
- [History View](#)
- [Tables View](#)



Note • The **Only Display View Nodes With Data** option on the **General** tab of the *ConflictSolver* and *Application Manager Options Dialog Box* controls whether product nodes (constituent views) appear if no data is contained in that view. If you select the option, products containing views without data will not display those views. For example, if a product has no **Dependencies**, then the **Dependencies** node is not displayed for that product.

Extended Attributes View

The Application Manager Extended Attributes View displays the optional extended attributes associated with the package. To open a package's Extended Attributes View, select the **Extended Attributes** node under that package in the package tree.

These attributes are dynamically populated based on an external [Extended Attribute Description File](#) (in XML format). You can specify the name and location of this file from the [Extended Attributes Tab](#) of the *ConflictSolver* and *Application Manager Options Dialog Box*.

On the left side of the view, the name for each attribute is displayed; the right side displays the value for the attribute. These values are in read-only fields, from which you can highlight and copy text, or, in the case of file links, launch the linked file. The file location displayed represents the location from which the file was originally imported; the file is actually stored within the Application Catalog and extracted into a temp directory when you click on the file link. This temp directory is purged when Application Manager closes, as long as the file is not locked by another application or process.

If you click on the attribute name, you can either provide the value in the [Group Properties Dialog Box](#) (for Text attributes) or browse for a file in a Browse dialog (for File attributes).



Note • You can also view and edit a package's extended attributes on the [Extended Attributes Page](#) of the ConflictSolver Process Assistant.

Files/Components View

When you expand a product in the Application Manager Product View, you can click on Files/Components to display the files and components in the product.

The following information is displayed:

Table 6-27 • File/Components View Information

Column	Description
Component	Name of component that the file listed in the FileName column is associated with.
FileName	Name of file.
FileSize	Size of the file listed in the FileName column.
Version	Version of the file listed in the FileName column.
Path	Installation location of the file listed in the FileName column.

INI File Changes View

When you expand a product in the Application Manager Product View, you can click on INI File Changes to display any INI file changes made by the product.

The following information is displayed:

Table 6-28 • INI File Changes View Information

Column	Description
Component	Name of component that makes an entry in the INI File.
FileName	Name of INI File that the component listed in the Component column makes an entry in.
DirProperty	The directory location where the INI File will be installed.
Section	The section of the INI file where this entry is made.
Key	The Key used in the INI File entry
Value	The Value used in the INI File entry.

Registry View

When you expand a product in the Application Manager Product View, you can click on Registry to display any registry entries created or changed by the product.

The following information is displayed:

Table 6-29 • Registry View Information

Column	Description
Component	The name of the component that is creating a Registry entry.
Root	Default value of Key.
Key	Key of the Registry Entry that this component is making.
Name	Name of the Registry Entry that this component is making.
Value	Value of the Registry Entry that this component is making.

Shortcuts View

When you expand a product in the Application Manager Product View, you can click on Shortcuts to display any shortcuts created by the product.

The following information is displayed:

Table 6-30 • Shortcuts View Information

Column	Description
Component	Name of the component that the shortcut listed in the Name column is associated with.
Name	Name of the shortcut.
Directory_	Directory where the shortcut will exist.
Target	Directory and executable that the shortcut invokes.

Merge Modules View

When you expand a product in the Application Manager Product View, you can click on Merge Modules to display any merge modules included the product.

The following information is displayed:

Table 6-31 • Merge Modules View Information


Column	Description
Title	The title of the Merge Module included with this package.
ModuleID	The number which uniquely identifies the Merge Module listed in the Title column.
Version	The version of the Merge Module listed in the Title column.
Language	The language that the Merge Module listed in the Title column was written for.

History View

When you expand a product in the Application Manager Product View, you can click on the **History** node to display a list of logged events for the selected package.

The following information is included:

Table 6-32 • Information Displayed in the History View

Item	Description
Action	<p>Name of the event which was logged:</p> <ul style="list-style-type: none"> • Import/Reimport • Validation • Conflict Detection • Conflict Resolution • Extended Attribute Modification • Package Description Modification • Package Move/Copy • Patch Analysis  <p>Note • You can specify which events you want to be listed on the History View by making selections on the History Tab of the ConflictSolver and Application Manager Options Dialog Box. You can choose which events you want to log and which events you want to display in the History View.</p>
Date	Date and time logged event occurred.
User	User who performed the logged event.
Description	Description providing details of the logged event.

Tables View

The Tables view provides a way to view the data for a given package in the Application Catalog. Select the specific table you want to view from the Tables list at the top of the view.

Most tables are derived directly from standard MSI tables, as described in the Windows Installer SDK online help.

When building your own ACE rules to use for conflict identification, it is important to understand the data available for packages so you can construct the necessary rule.

OS Snapshot View

When you click on an OS Snapshot in the Application Manager Product View, details about the snapshot appear in the right pane of the user interface.

The following information is displayed:

Table 6-33 • OS Snapshot View Information

Field	Description
Version	Version of the operating system of the OS Snapshot, such as Windows XP - 5.1.2600.
Language	The language the operating system was written for.
File	This can be either a hard-coded path or a UNC path.
Imported On	The date and time the OS Snapshot was imported.
Description	You can edit this with additional information about the OS Snapshot.

Click the plus sign next to the OS Snapshot icon to view these OS Snapshot constituent views:

- [Extended Attributes View for OS Snapshots](#)
- [Files View for OS Snapshots](#)
- [INI File Changes View for OS Snapshots](#)
- [Registry View for OS Snapshots](#)
- [Shortcuts View for OS Snapshots](#)

Extended Attributes View for OS Snapshots

The Extended Attributes View for an OS Snapshot is identical to the Extended Attributes View shown when a Product is selected. See [Extended Attributes View](#) in the Application Manager Product View section.

Files View for OS Snapshots

When you expand an OS Snapshot in the Application Manager Product View, you can click on Files to display a list of the files included in the OS Snapshot.

The following information is displayed for each of the files included in the OS Snapshot:

Table 6-34 • Files View Information

Column	Description
FileName	Name of the file.

Table 6-34 • Files View Information

Column	Description
csFilePath	Path
FileSize	Size of the OS Snapshot file.
Version	Version of the OS Snapshot file.
Language	Language that the OS Snapshot file was written for.
Attributes	Any attributes associated with the file.

INI File Changes View for OS Snapshots

When you expand an OS Snapshot in the Application Manager Product View, you can click on INI File Changes to display any INI file changes made by the snapshot.

The following information is displayed for each change to the INI file that is made by the snapshot:

Table 6-35 • INI File Changes View Information

Column	Description
FileName	Name of INI File that the OS Snapshot makes an entry in.
csFilePath	Path
Section	The section of the INI File where this entry is made.
Key	The Key used in the INI File entry
Value	The Value used in the INI File entry.

Registry View for OS Snapshots

When you expand an OS Snapshot in the Application Manager Product View, you can click on **Registry** to display any registry entries created or changed by the product.

The following information is displayed for each Registry Entry:

Table 6-36 • Registry View Information

Column	Description
Root	Default value of Key.
Key	Key of the Registry Entry that this component is making.

Table 6-36 • Registry View Information

Column	Description
Name	Name of the Registry Entry that this component is making.
Value	Value of the Registry Entry that this component is making.

Shortcuts View for OS Snapshots

When you expand an OS Snapshot in the Application Manager Product View, you can click on Shortcuts to display any shortcuts that were found on this OS Snapshot's operating system.

The following information is displayed:

Table 6-37 • Shortcuts View Information

Column	Description
Name	Name of shortcut.
Directory	Location of shortcut.
Target	The application executable that the shortcut points to.

Tables View for OS Snapshots

The Tables View for an OS Snapshot is identical to the Tables View shown when a Product is selected. See [Tables View](#) in the Application Manager Product View section.

Other Setup Types View

When you click on an Other Setup Types package (a package containing non-MSI based setup files) in the Application Manager Product View, details about that setup are displayed in the right pane of the user interface.

The following information is included:

Table 6-38 • Other Setup Types View Information

Field	Description
Main Directory	The original location of the non-MSI based setup files when they were imported into the Application Catalog.
Imported On	The date and time the setup was imported.

Table 6-38 • Other Setup Types View Information

Field	Description
Files	<p>A listing of the individual files making up the setup are listed, allowing you to view or delete files by making a selection from the context menu.</p> <ul style="list-style-type: none"> To view the contents of a file, select View from the context menu. You can only view files supported by applications installed on your workstation. If you attempt to view a non-supported file, no View window will appear. If you attempt to view an executable, that executable will be launched. To delete a file from this setup, select Delete from the context menu.

If you expand an Other Setup Type, you can view its constituent view, the [Extended Attributes View for Marimba NCP Files](#).

Extended Attributes View for Other Setup Types

The Extended Attributes View for an Other Setup Types package is identical to the Extended Attributes View shown when a Product is selected. See [Extended Attributes View](#) in the Application Manager Product View section.

Marimba NCP Files Views



Edition • Import support for Marimba Native Channel Packager (.ncp) files is available in AdminStudio Enterprise Edition.

When you select a Marimba NCP file (.ncp) from the Application Manager Product View, details about that file are displayed in the right pane of the user interface.

The following information is included:

Table 6-39 • NCP Views Information

Column	Description
Conflict Results	Date that this package was last checked for conflicts. Click the link to jump to the Conflicts View for Marimba NCP Files .
Version	Version of the software package
File	Name and path of the NCP file that was imported
Imported On	Date when the file was imported
Description	After the NCP file is imported, you can edit this with additional information.

If you expand an NCP file, you can view its constituent views:

- [Extended Attributes View for Marimba NCP Files](#)
- [Files View for Marimba NCP Files](#)
- [INI File Changes View for Marimba NCP Files](#)
- [Registry View for Marimba NCP Files](#)
- [Shortcuts View for Marimba NCP Files](#)
- [Tables View for Marimba NCP Files](#)

Extended Attributes View for Marimba NCP Files

The Extended Attributes View for an NCP File is identical to the Extended Attributes View shown when a Product is selected. See [Extended Attributes View](#) in the Application Manager Product View section.

Files View for Marimba NCP Files

When you expand a Marimba NCP File in the Application Manager Product View, you can click on Files to display the files included in the NCP File.

The following information is displayed for each file:

Table 6-40 • Files View Information

Column	Description
FileName	Name of the file.
csFilePath	Path
FileSize	Size of the NCP file.
Version	Version of the NCP file.
Language	Language that the NCP file was written for.
Attributes	Any attributes associated with the file.

INI File Changes View for Marimba NCP Files

When you expand an Marimba NCP File in the Application Manager Product View, you can click on INI File Changes to display any INI file changes made by the NCP file.

The following information is displayed for each change that is made to the INI file by the NCP file:

Table 6-41 • INI File Changes View Information

Column	Description
FileName	Name of the file.
csFullPath	Path
Section	Section of INI file that is changed.
Key	Key that is changed.
Value	Value that is changed.

Registry View for Marimba NCP Files

When you expand a Marimba NCP File in the Application Manager Product View, you can click on Registry to display any registry entries created or changed by the NCP file.

The following information is displayed for each registry entry:

Table 6-42 • Registry View Information

Column	Description
Root	Default value of Key.
Key	Key of the Registry Entry that this NCP file is making.
Name	Name of the Registry Entry that this NCP file is making.
Value	Value of the Registry Entry that this NCP file is making.

Shortcuts View for Marimba NCP Files

When you expand a Marimba NCP file in the Application Manager Product View, you can click on Shortcuts to display any shortcuts created by the NCP file.

The following information is displayed for each shortcut created by the NCP file:

Table 6-43 • Shortcuts View Information

Field	Description
Name	Name of the shortcut.
Directory	Directory where the shortcut will exist.
Target	Directory and executable that the shortcut invokes.

Tables View for Marimba NCP Files

The Tables View for an NCP file is identical to the Tables View shown when a Product is selected. See [Tables View](#) in the Application Manager Product View section.

Virtual Package View

When you select a virtual package in the Application Catalog, the Virtual Package view opens.

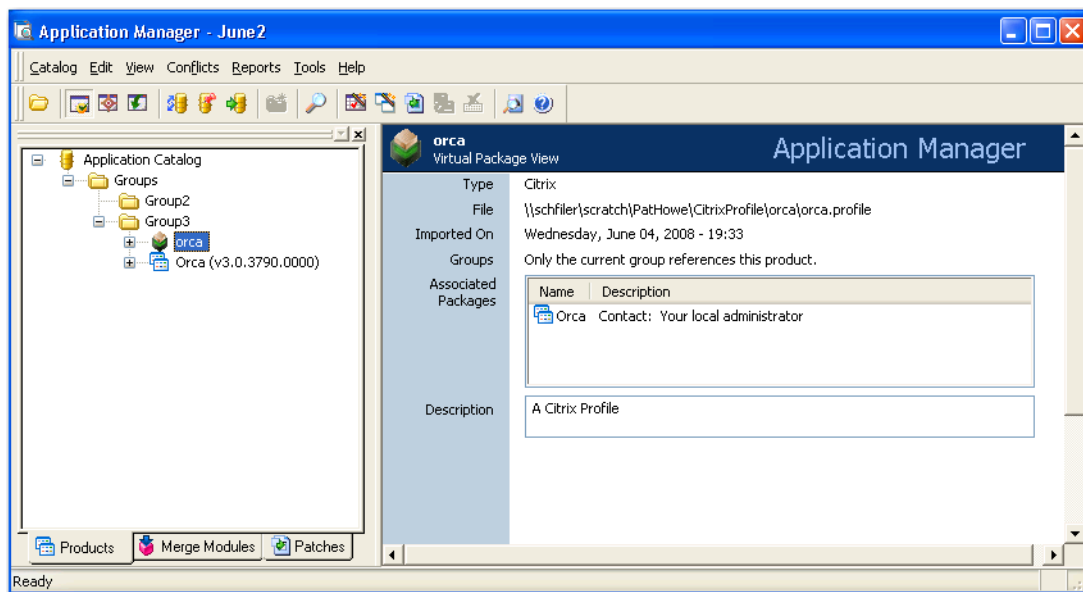



Figure 6-7: Virtual Package View

The following information is displayed:

Table 6-44 • Virtual Package View

Option	Description
Type	Identifies the type of virtual package. Options are: <ul style="list-style-type: none"> • Citrix—A Citrix profile. • ThinApp—A ThinApp application.
File	Identifies the location of this package. It can be either a hard-coded path or a UNC path. If the package is part of the Software Repository, the following statement appears: Managed within the Software Repository
Imported On	Date and time virtual package was imported into the Application Catalog.
Groups	Lists all any groups to which the virtual package belongs.
Associated Packages	<p>Lists the virtual package's source Windows Installer package.</p> <p>By associating a virtual package with the Windows Installer package which originated it, you have the convenience of being able to easily locate the virtual package's originating Windows Installer package, modify the original Windows Installer package, and then regenerate the virtual package.</p>  <p>Note • To quickly locate one of these listed Windows Installer packages in the Application Catalog, double-click on it.</p> <p>In order to be listed here, the Windows Installer package has to have already been imported into the Application Catalog and must be associated with this virtual package. There are several ways to associate a virtual package with its source Windows Installer package:</p> <ul style="list-style-type: none"> • Importing a Virtual Package During the Import of its Source Windows Installer Package • Importing a Virtual Package After the Import of its Source Windows Installer Package • Manually Associating a Virtual Package with a Windows Installer Package
Description	You can edit this field and add additional information about the virtual package.

Patches Tab Views

The **Patches** tab includes the following views which provide content and impact information about Microsoft patches that have been imported into the Application Catalog.

Table 6-45 • Patches Tab Views

View	Description
Patches Group View	Opens when the root group in the Patches tab is selected, and lists all of the groups that have been created in the Patches tab.
New Patches Group View	All new patches are imported into the New Patches group, and this view lists all of the patches in that group.
Group View of a Selected Group	Opens when a group other than the root group in the Patches tab is selected. For each selected group, a list of all of the patches in that group is displayed.
Patch View	Lists general content information on a selected patch.

Patches Group View

The Patches Group View opens when the root group in the **Patches** tab is selected. The Patches Group View lists all groups that have been created in the Patches tab.

All new patches are imported into the **New Patches** group, and then you can organize the patches into other groups according to your business needs. See [Organizing Your Application Catalog Using Groups](#).

The **New Patches** group is automatically created during installation. While it can be renamed, it cannot be deleted.

Context Menu Options

When the root group in the Patches Tab is selected, the following items are available on the context menu:

- **Refresh**—Refresh the patch listing to reflect the most recent modifications.
- **Import Patches**—Opens the OS Security Patch Wizard so that you can import patches into the Application Catalog.
- **New Group**—Create a new group.
- **Rename**—Rename the selected group.
- **Properties**—Open the **Group Properties** dialog box.

New Patches Group View

The New Patches Group View opens when the **New Patches** group on the Application Manager or ConflictSolver **Patches** tab is selected. All new patches are imported into the **New Patches** group. You can then organize the patches into other groups according to your business needs. See [Organizing Your Application Catalog Using Groups](#).

The **New Patches** group is automatically created during installation. While it can be renamed, it cannot be deleted.

The New Patches Group View displays a list of all of the patches in that group, including the following information:

Table 6-46 • New Patches Group View Information

Option	Description
Name	Name of patch file.
Description	Description of the patch file.
Release Date	Date the patch was released by Microsoft.
Import Date	Date the patch was imported into the Application Catalog.

If you select a patch in this list, detailed patch properties are displayed in the area to the left of the list, including the **Title**, **Summary**, and **Release Date** of the patch.

Context Menu Options

When the **New Patches** group in the **Patches Tab** is selected, the following items are available on the context menu:

- **New Group**—Create a new group.
- **Rename**—Rename the selected group.
- **Properties**—Open the **Group Properties** dialog box.

Group View of a Selected Group

The Group View of a selected group opens when a group other than the root group or the **New Patches** group in the **Patches** tab is selected. For each selected group, a list of all of the patches in that group is displayed, including the following information:

Table 6-47 • Group View of a Selected Group Information

Option	Description
Name	Name of patch file.
Description	Description of the patch file.
Release Date	Date the patch was released by Microsoft.
Import Date	Date the patch was imported into the Application Catalog.

If you select a patch in this list, detailed patch properties are displayed in the area to the left of the list, including the **Title**, **Summary**, and **Release Date** of the patch.

Context Menu Options

When a group other than the root group or the **New Patches** group in the **Patches Tab** is selected, the following items are available on the context menu:

- **New Group**—Create a new group.
- **Rename**—Rename the selected group.
- **Delete**—Delete the selected group.
- **Cut**—Copy the selected group to the clipboard so that you can remove it from its current location and paste it into another location.
- **Paste**—Insert the patch or the group on the clipboard into the selected location.
- **Properties**—Open the **Group Properties** dialog box.

Patch View

The **Patch View**, which is displayed when a patch is selected on the **Patches** tab, lists general information on the selected patch. The following information is included:

Table 6-48 • Patch View Information

Option	Description
ID	Microsoft Security Bulletin ID. Click this link to view this bulletin on the Microsoft Web site.
Title	Title of patch.

Table 6-48 • Patch View Information

Option	Description
Release Date	Date the patch was released by Microsoft.
KB Article	Microsoft Knowledge Base article ID. Click this link to view this article on the Microsoft Web site.
Imported On	Date patch was imported into the Application Catalog
Groups	List of all of the groups that this patch is included in.
Description	You can enter a description of the patch in this field.

In the Application Manager, you can view additional detailed patch information by selecting a patch and then selecting **Properties** from the context menu.

Context Menu Options

When a patch in the **Patches Tab** is selected, the following items are available on the context menu:

- **Rename**—Rename the selected patch.
- **Delete**—Delete the selected patch.
- **Cut**—Copy the selected patch to the clipboard so that you can remove it from its current location and paste it into another location.
- **Copy**—Copy the selected patch to the clipboard so that you can paste a copy of it into another location.
- **Paste**—Insert the patch on the clipboard into the selected location.
- **Generate Report**—Generate a Patch Impact Analysis Report for that patch. See [Generating the Patch Report](#).
- **Properties**—Open the **Patch Properties** dialog box for that patch.

Merge Module Views

From the Merge Modules tab, you can access the following views:

- **All Merge Modules View**—Select this option to view a list of all of the Merge Modules in your catalog. See [All Merge Modules View](#).
- **Merge Module View**—Select an individual Merge Module to see detailed information on that module. See [Merge Module View](#).

All Merge Modules View

The All Merge Modules view is the root node of the merge modules explorer. It contains a list of all merge modules in the Application Catalog, including titles, versions, languages, and identifiers for each module.

Double-click on a merge module to see information about it in the [Merge Module View](#).

Merge Module View

When you select a merge module in the merge modules explorer, details are displayed in this view.

These details include:

Table 6-49 • Merge Module View Information

Field	Description
Version	The version of the Merge Module.
Language	The language that the Merge Module was written for.
Identifier	String that uniquely identifies the Merge Module.
File	The path and file name of the Merge Module file that was imported.
Imported On	The date and time the Merge Module was imported.

Click the plus sign next to the Merge Module to view these Merge Module constituent views:

- [Merge Module / Components View](#)
- [Merge Module / Dependency View](#)
- [Merge Module / Exclusion View](#)
- [Merge Module / Files View](#)
- [Merge Module / Products View](#)

Merge Module / Components View

When you expand a merge module in the merge modules explorer, you can click on Components to display any components created or changed by the merge module.

The following information is displayed for each component included in this merge module:

- Component
- ComponentId
- Directory_
- csFullPath

Merge Module / Dependency View

When you expand a merge module in the merge modules explorer, you can click on Dependency to display any dependencies in the merge module.

The following information is displayed for each dependency included in this merge module:

- ModuleLanguage
- RequiredID
- RequiredLanguage
- RequiredVersion

Merge Module / Exclusion View

When you expand a merge module in the merge modules explorer, you can click on Exclusion to display any exclusions in the merge module.

The following information is displayed for each exclusion included in this merge module:

- ModuleLanguage
- ExcludedID
- ExcludedLanguage
- ExcludedMaxVersion
- ExcludedMinVersion

Merge Module / Files View

When you expand a merge module in the merge modules explorer, you can click on Files to display any files in the merge module.

The following information is displayed:

Table 6-50 • Merge Module Files View Information

Column	Description
Component_	Name of component that this Merge Module file is associated with.
File	Name of this Merge Module file.
FileName	File name of this Merge Module file.
FileSize	Size of this Merge Module file.
Version	Version of this Merge Module file.

Merge Module / Products View

When you expand a merge module in the merge modules explorer, you can click on Products to display any products in the Application Catalog that use the merge module.

The following information is displayed:

Table 6-51 • Merge Module Products View Information

Column	Description
ProductName	Name of product associated with this Merge Module.
ProductVersion	Version of product associated with this Merge Module.
Manufacturer	Manufacturer of the product associated with this Merge Module.

Dialog Boxes

Application Manager includes the following dialog boxes:

- [Application Catalog Properties Dialog Box](#)
- [Associate with Workflow Manager Application Dialog Box](#)
- [Command-Line Parameters Dialog Box](#)
- [Command-Line Parameters Dialog Box](#)

- [Connect Application Catalog Dialog Box](#)
- [Extended Attribute Property Dialog Box](#)
- [Find Dialog Box](#)
- [Login Required Dialog Box](#)
- [Group Properties Dialog Box](#)
- [Package Auto Import Dialog Box](#)
- [Publication Manager Dialog Box](#)
- [Publication Processing Dialog Box](#)
- [Publication Properties Dialog Box](#)
- [Select AdminStudio Enterprise Server URL Dialog Box](#)
- [Subscription Manager Dialog Box](#)
- [Task Scheduling Dialog Box](#)

Application Catalog Properties Dialog Box

The Application Catalog Properties Dialog box, which is opened by selecting **Properties** on the **Catalog** menu, is used to enable the Software Repository, select its location, and create a Proxy Account that AdminStudio can use to make modifications to the Software Repository.

Software Repository Tab

A Windows Installer package is made up of many files that are executed when the setup is run. You only import the .msi file into the Application Catalog, not all of the files necessary for installation. With the Software Repository, when you import an installation package into the Application Catalog, all of the files associated with that package are copied into the Software Repository location, a directory that you specify. This allows you to manage those files, preventing them from getting modified or lost.

On the **Software Repository tab**, you can choose to **Enable the Software Repository** for the new Application Catalog, and specify a **Proxy Account** for AdminStudio to use to make modifications to the directory path selected as the **Software Repository Location**.

This tab includes the following options:

Table 6-52 • Application Catalog Properties / Software Repository Tab Options

Option	Description
Enable Software Repository	Select this option to enable the Software Repository feature for this Application Catalog.
Software Repository Location	Specify the directory location of the Software Repository for this Application Catalog.

Table 6-52 • Application Catalog Properties / Software Repository Tab Options (cont.)

Option	Description
Proxy Account	Specify a Login ID and Password for AdminStudio to use when modifying the Software Repository Location directory.

Proxy Account Tab

For SQL Server Application Catalogs, you need to specify a Proxy Account for AdminStudio to use to access the Application Catalog on the database server. You cannot use Windows Authentication for this Proxy Account.

Enter a valid Login ID and Password to log on to the database server that contains this Application Catalog.

Associate with Workflow Manager Application Dialog Box



Note • AdminStudio Workflow Manager is a Web-based application management system that has integrated functionality with AdminStudio.

The Associate with Workflow Manager Application dialog box is displayed when you select the corresponding command from the context menu when right-clicking on a product. It allows you to associate extended attribute data for a product with an application already in an AdminStudio Workflow Manager application catalog. You can select the available packages from the list in the dialog box.



Note • This dialog (and its corresponding command) are only available if you select the **Integrate with Workflow Manager** option on the **Extended Attributes Tab** of the ConflictSolver and Application Manager **Options Dialog Box**.



Note • Be sure to select a Workflow Manager application that uses a template containing at least one major data group that was specified with the group's extended attribute description file, as described in the AdminStudio Workflow Manager user documentation.

Change Enterprise Server Password Dialog Box

This dialog box, which is opened by selecting **Change AES Password** on the **Catalog** menu, allows you to change your password to connect to the AdminStudio Enterprise Server Application Catalog (which is the same password you use to log in to the AdminStudio Enterprise Server). The **Change AES Password** selection is enabled when you are connected to the AdminStudio Enterprise Server Application Catalog.



Note • If you are not connected to the AdminStudio Enterprise Server Application Catalog, the **Change AES Password** selection on the **Catalog** menu is disabled.

Table 6-53 • Change Enterprise Server Password Dialog Box

Option	Description
User Name	(Read Only) User name of user who is connected to the AdminStudio Enterprise Server Application Catalog.
Old Password	Enter the existing password for current user.
New Password	Enter the new password.
Confirm New Password	Enter the new password a second time.

Command-Line Parameters Dialog Box

This dialog box is displayed when running Application Manager from the command line using the `-?` parameter. Information in this dialog box is covered in the [Application Manager Command-Line Functionality](#) topic.

Connect Application Catalog Dialog Box

The Connect Application Catalog dialog box opens when you choose to open an existing Application Catalog. This dialog box has three tabs:

- **Enterprise Server**—Select this tab to open the AdminStudio Enterprise Server Application Catalog database. See [Enterprise Server Tab](#).
- **Standalone**—Select this tab to open an Application Catalog database other than the AdminStudio Enterprise Server Application Catalog. See [Standalone Tab](#).
- **Recent**—Provides a list of recently opened Application Catalogs. When you select an Application Catalog and click **OK**, either the Application Catalog opens or you are prompted for login information (if you need authentication to the Application Catalog). See [Recent Tab](#).

Making this the Default Shared Application Catalog

If you select the **Make this the default shared Application Catalog** option, the Application Catalog you are opening will become the default Application Catalog (and be recorded as such in the **AdminStudio Shared** directory).

If the Application Catalog is made the default, all other AdminStudio users that use the same shared directory will automatically connect to the default Application Catalog when AdminStudio is launched. Therefore, you should only set this option if you want to affect all AdminStudio users who access that shared directory.




Note • In the AdminStudio Enterprise Edition, only the AdminStudio Administrator or users with the Change Default Database permission will see the **Make this the default shared Application Catalog** option. This allows the AdminStudio Administrator to configure the default Application Catalog, and then subsequent installations of AdminStudio will automatically connect to the default Application Catalog if they use the same shared directory.

Enterprise Server Tab

To connect to the AdminStudio Enterprise Server Application Catalog, you log in on the **Enterprise Server** tab of the **Connect Application Catalog** dialog box.

Table 6-54 • Connect Application Catalog / Enterprise Server Tab Options

Option	Description
URL	<p>Location of the AdminStudio Enterprise Server associated with this installation of AdminStudio.</p> <p>If the AdminStudio Enterprise Server has not yet been configured with the AdminStudio client tools (such as when it is set to its default value of <code>http://localhost</code>), click the URL link to open the Select AdminStudio Enterprise Server URL dialog box, and enter the URL for location of the AdminStudio Enterprise Server associated with this installation of AdminStudio.</p>
Authentication	<p>Select one of the following options:</p> <ul style="list-style-type: none"> Windows Authentication AdminStudio Enterprise Server User  <p>Note • When using AdminStudio Enterprise Server User authentication, if Anonymous authentication is turned off in IIS, both the user's machine and the AdminStudio Enterprise Server need to be on the same domain in order for login to succeed.</p>
User Name and Password	<p>If you selected AdminStudio Enterprise Server User, enter your AdminStudio Enterprise Server User Name and Password (provided by your System Administrator).</p>

Login Troubleshooting

If you are using a Web Portal with custom security zone settings, your AdminStudio Enterprise Server URL is using an IP address, and you receive Error 0x800A1518 when you attempt to login, change the AdminStudio Enterprise Server URL to the NetBios equivalent and then try again. For example, if you are connecting to `http://120.12.1.15`, the NetBios equivalent would be `http://wfmpor1a1`.



Standalone Tab

On the **Standalone** tab of the **Connect Application Catalog** dialog box and the **Specify Database Information** panel of the [Application Catalog Wizard](#), enter the information required to login to the specified Application Catalog or enter the name of the Application Catalog that you are creating.

Table 6-55 • Connect Application Catalog / Standalone Tab Options

Option	Description
Server	Select one of the available SQL Servers on the network from this list. You can also manually enter the name of the SQL Server to which you want to connect.
Authentication	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Windows Authentication—Choose to use Windows network authentication (your network login ID) to log into this Application Catalog. • Server Authentication—Choose to use SQL Server login identification for authentication. • Login ID and Password—If you chose Server Authentication, enter the appropriate Login ID and Password.
Catalog	<p>Do one of the following:</p> <ul style="list-style-type: none"> • If you are connecting to an existing Application Catalog, select the catalog from those available on the Server. • If you are creating a new Application Catalog, enter a name for this new catalog.
Test	Click this button to test whether a connection can be made to the database.
<p>Make this the default shared Application Catalog</p> <p><i>Connect Application Catalog Dialog Box Only</i></p>	<p>When this option is selected, the Application Catalog you are trying to open or create will become the default Application Catalog (and be recorded as such in the AdminStudio Shared directory).</p> <p>If the Application Catalog is made the default, all other AdminStudio users that use the same shared directory will automatically connect to the default Application Catalog when AdminStudio is launched. Therefore, you should only set this option if you want to affect all AdminStudio users who access that shared directory.</p>

Table 6-55 • Connect Application Catalog / Standalone Tab Options (cont.)

Option	Description
Create database as a subscriber <i>Create Application Catalog Wizard Only</i>	<p>Select this check box to designate this new Application Catalog as a Subscriber database in Application Catalog Replication (a database that receives replicated data). When this option is not selected, or when an existing database is upgraded, the Application Catalog is defined as a Publisher database in Application Catalog Replication (a database that makes data available for replication).</p> <p>The AdminStudio interface selectively enables/disables the Replication menu items on the Catalog menu based upon the type of Application Catalog that is currently opened.</p>  <p>Edition • Application Catalog Replication is included with the AdminStudio Enterprise Edition. If you have AdminStudio Standard or Professional Editions, this check box is disabled. Contact your AdminStudio Sales Representative for more information.</p>  <p>Note • This check box only appears when creating a new Application Catalog, not when logging in to an existing Application Catalog.</p>

Recent Tab

The **Recent** tab displays a list of all Application Catalogs that have recently been open. To login to one of these Application Catalogs, select it and click OK.

- If you are opening a standalone SQL Server Application Catalog, you will be prompted for login information.
- If you are opening the AdminStudio Enterprise Server Application Catalog, you are prompted for AdminStudio Enterprise Server login information before the Application Catalog will open.

Default Application Catalog Dialog Box

When you initially open AdminStudio, because a default Application Catalog has not yet been set, the **Default Application Catalog** dialog box opens, prompting you to open an Application Catalog.

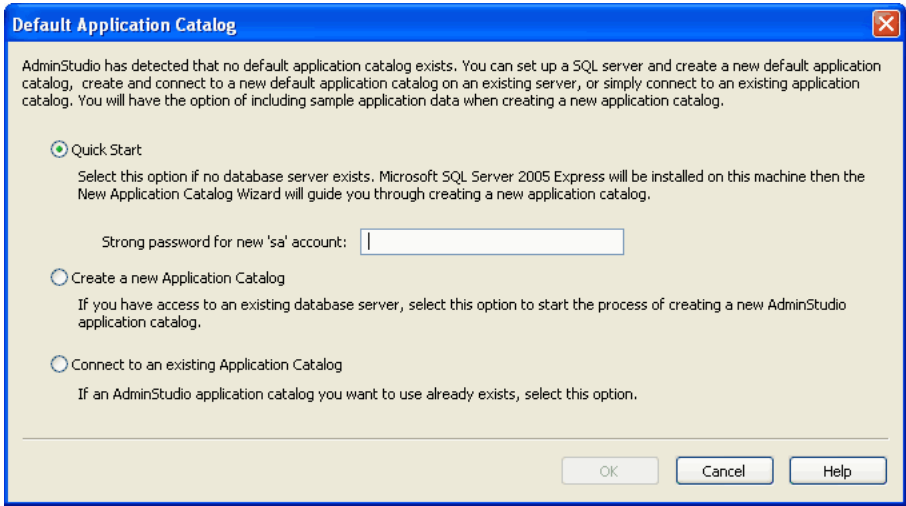


Figure 6-8: Default Application Catalog dialog box

You can select any of the following options:

Table 6-56 • Default Application Catalog Dialog Box Options

Option	Description
Quick Start	<p>Select this option if you do not have access to a database server. Microsoft SQL Server 2005 Express will be installed on your machine first, then you will have the option to create a new Application Catalog database on that server that is populated with sample data.</p> <p>When Microsoft SQL Server 2005 Express is installed, a user named sa is automatically created. Therefore, you need to enter a password for this account in the Strong password for new 'sa' account box. A strong password:</p> <ul style="list-style-type: none">• must be at least 7 characters long,• must be a combination of letters, numbers, and symbol characters,• must not be a dictionary word, a command name, a person’s name, or a system user name• must not contain any of the following characters: [] { } () , ; ? * ! @

Table 6-56 • Default Application Catalog Dialog Box Options

Option	Description
Create a new Application Catalog	Select this option to create a new, empty Application Catalog on an existing SQL Server database server that you have access to. You will have the option of populating this new Application Catalog with sample data.
Connect to an existing Application Catalog	Select this option to connect to an existing Application Catalog on an SQL Server database server.

Extended Attribute Property Dialog Box

If you use extended attributes, and you click on a text extended attribute label, this dialog box opens.

Within it, you can provide the value for the extended attribute. When you click OK, the value is automatically displayed in the [Extended Attributes View](#) next to the corresponding label.

Find Dialog Box

You can use the **Find** dialog box, which is accessed by selecting **Find** from the **Edit** menu, to search for data in Application Catalog tables.



Note • This search is limited to string type columns.


The tables that are searched depend upon what is selected when the **Find** dialog box is opened:

Table 6-57 • Search Options

If you select...	and specify these options...	this will be searched
Group	All Tables and All Columns	All tables and all columns in all of the Packages in the selected Group
Package	All Tables and All Columns	All tables and all columns in the selected Package
Package	A Table and All Columns	All columns of a specific table in the selected Package
Package	A Table and a Column	A specific column in a specific table in the selected Package

Also, if you want to search for a partial match rather than an exact match, you can use the Partial Match option on the Find dialog box.

The Find dialog box can be accessed in four ways:

- Select **Find** on the **Edit** menu.
- Press **Ctrl + F**.
- Click the **Find** button () on the toolbar.
- Select the node in the tree (Group, MSI Package, OS Snapshot, Other Setup, etc.) that you want to search, and select **Find** from the context menu.

The **Find** dialog box contains the following fields and buttons:

Table 6-58 • Find Dialog Box Properties

Properties	Description
Find What	Enter the text that you want to search for.
Look In Table	<ul style="list-style-type: none"> • If a package was selected when the Find dialog box was opened, all of the tables in that package are listed. Select the table that you would like to search, or select <All Tables>. When you select a table from this list, the Look In Columns list is populated with all of the columns in that table. • If a group is selected when you opened the Find dialog box, <All Tables> is the only option listed.
Look In Columns	<ul style="list-style-type: none"> • If you selected a table from the Look in Table list, all of the columns in that table are listed. Select the column that you would like to search, or select <All Columns>. • If a group was selected when you opened the Find dialog box, <All Columns> is the only option listed.
Partial Match	<ul style="list-style-type: none"> • If this option is not selected, Application Manager will search for an exact match of the text you entered in the Find What text box. The search will be case sensitive. • If this option is selected, then Application Manager will use appropriate wild card characters so that a partial data match is performed. The search will be case insensitive.

Table 6-58 • Find Dialog Box Properties

Properties	Description
Find Button	<p>Click to initiate the search. The Find dialog box will close, and the data that is found is displayed in the Search Results tab of the Output Window, in the following format:</p> <div></div> <p>If you double click on this data, Application Manager will navigate to the appropriate record in the Tables View, and that record will be highlighted in red.</p>

Login Required Dialog Box

If you chose to open an existing Application Catalog that was listed on the **Recent** tab of the [Connect Application Catalog Dialog Box](#), and you are not currently logged in to that database, this dialog box opens prompting you to log in.

Group Properties Dialog Box

The Group Properties dialog box opens when you right-click a group in the Groups tree and select Properties from the context menu.

You can provide both a name and description for the group, as well as any comments. Information provided in this dialog box is displayed in the Properties area of the Group view (displayed whenever a group is selected in the main Application Manager Product View).

Table 6-59 • Group Properties Dialog Box Properties

Option	Description
Name	Provide a name for the group. This name cannot exceed 40 characters.
Description	Enter any descriptive information about the group. This cannot exceed 80 characters.
Comments	Provide any comments about the group. Comments cannot exceed 255 characters in length.

Package Auto Import Dialog Box



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

The Package Auto Import dialog box lists the Application Catalogs and Network Directories that are being monitored.

The Package Auto Import dialog box includes the following options:

Table 6-60 • Package Auto Import Dialog Box Options


Option	Description
Monitored Activities	<p>List of all Remote Application Catalogs and Network Directories that are being monitored for new or updated packages. Only those items linked to the currently open Application Catalog will be listed.</p>  <p>Note • When you initially set up a connection to a Remote Application Catalog, until the linked packages have been imported into the local Application Catalog for the first time, the Start button is disabled when that link is selected on the Package Auto Import dialog box. The first automatic import of linked packages in a Remote Application Catalog is initiated when you click the Close button to exit the Package Auto Import dialog box.</p>
Occurrence	Lists how often the Application Catalog or Network Directory is going to be checked for updated or new packages. This corresponds to the current selection made on the Options tab of the Package Auto Import Properties dialog box for that item.
Update On	Lists the last time the Remote Application Catalog or Network Directory was checked for new or updated packages.
Add	Click to access the Package Auto Import Wizard , where you can set up a connection to a Remote Application Catalog or Network Directory.
Remove	Click to remove the selected Remote Application Catalog or Network Directory connection. The packages that have already been imported into your Application Catalog will remain, but they will no longer be automatically updated.
Properties	Click to access the Package Auto Import Properties (Remote Application Catalog) Dialog Box or the Package Auto Import Properties (Network Directory) Dialog Box , where you can change the update schedule interval and other options for the selected link. This button is disabled for Remote Application Catalog entries until the linked packages have been imported into the Application Catalog for the first time.

Table 6-60 • Package Auto Import Dialog Box Options (cont.)

Option	Description
Start	Click to force an update of the selected link. Package Auto Import checks the monitored Remote Application Catalog or Network Directory to see if there were any updated or new packages, and then initiates an auto-import of those packages. This button is disabled for Remote Application Catalog entries until the linked packages have been imported into the Application Catalog for the first time.
Close	Click to exit this dialog box. When you initially set up a connection to a Remote Application Catalog or Network Directory, automatic import will be initiated when you click the Close button.



Note • If you first use the Package Auto Import Wizard to set up a connection to a package, and then use the Package Auto Import Wizard to set up a connection to another package in the same Application Catalog, those connections will be listed as two separate entries on the Package Auto Import dialog box. However, after the packages have been imported into the Application Catalog, only one entry will appear for that Application Catalog.

Package Auto Import Properties (Remote Application Catalog) Dialog Box



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

The Package Auto Import Properties (Remote Application Catalog) dialog box is accessed by selecting an Application Catalog on the [Package Auto Import Dialog Box](#) and clicking Properties.



Note • See also the [Package Auto Import Properties \(Network Directory\) Dialog Box](#).

Groups Tab

The Groups tab displays the packages within a Remote Application Catalog that you can link to. Select or deselect the check boxes to link/unlink to the packages. Any package that is already linked to will display the check box as selected.

Options Tab

The settings on the Options tab directly affect the way linked packages will be refreshed (update the contents of packages on your local Application Catalog with packages on Remote Application Catalogs).

Table 6-61 • Options Tab Properties

Properties	Description
Update package when its source is modified	<p>If you have selected this option:</p> <p>When the scheduled automatic import occurs or when you select Start on the Package Auto Import dialog box, Application Manager will go and check if any of the linked packages have been modified in the Application Catalog.</p> <ul style="list-style-type: none"> • If the linked package has been modified, then the local Application Catalog database is updated to reflect the changes. • If the linked package has not been modified, you receive a message stating that no refresh is necessary. <p>If you have not selected this option:</p> <p>When the scheduled automatic import occurs or when you select Start on the Package Auto Import dialog box, Application Manager will still check to see if any of the linked packages have been modified:</p> <ul style="list-style-type: none"> • If the linked package has been modified, Application Manager will inform you that the source package has been modified but that it will not update the local Application Catalog database because this option is not selected. • If the linked package has not been modified, you receive a message stating that no refresh is necessary.
Delete package when the source package is deleted	<p>If you have selected this option, when the scheduled automatic import occurs or when you select Start on the Package Auto Import dialog box, Application Manager will delete any linked data in the local database if the data is deleted from the remote database.</p> <p>If you have not selected this option, when the scheduled automatic import occurs or when you select Start on the Package Auto Import dialog box, even if the source package is deleted in the remote database, your copy of the package will not be deleted.</p>
Schedule	Select this option to enable automatic import of the linked packages at scheduled intervals. To disable automatic import, leave this option unselected. When this option is selected, the scheduling fields are enabled.
Run the Process	From this list, select Daily or Weekly to specify how often you want automatic import to be performed.

Table 6-61 • Options Tab Properties (cont.)

Properties	Description
At	Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
Starting on	Click on the arrow to open the Calendar and then select the first day that you want automatic importing to begin.

Package Auto Import Properties (Network Directory) Dialog Box



Note • To use the Package Auto Import feature, you must have AdminStudio Administrator permission. For all other users, the Package Auto Import options will be disabled.

The Package Auto Import Properties (Network Directory) dialog box is accessed by selecting a network directory on the [Package Auto Import Dialog Box](#) and clicking Properties. On this dialog you can choose to monitor sub-directories of the selected directory, and can schedule the monitor interval for this directory.



Note • See also the [Package Auto Import Properties \(Remote Application Catalog\) Dialog Box](#).

This dialog box contains the following options:

Table 6-62 • Package Auto Import Properties (Remote Catalog)

Option	Description
Folder	Displays a read-only view of the network directory that is being monitored.
Monitor sub-directories also	Select this option if you want to check sub-directories of the selected directory for Windows Installer packages to import.
Schedule	<p>Select this option to turn on automatic monitoring of the selected directory. If you do not select this option, the network directory will only be monitored when Start is clicked on the Package Auto Import dialog box. When the Schedule option is selected, the following fields are activated, where you can specify the monitor interval for this directory.</p> <ul style="list-style-type: none"> • Run the process—Select whether you want to run the monitoring process Daily or Weekly. • At—Specify the time that you want to run the monitoring process. • Starting on—Select the initial day that you want to begin the monitoring process.

Publication Manager Dialog Box

The Publication Manager lists all Publications in the SQL Server database you are currently connected to. Use this dialog box to create, edit, publish, or delete a Publication.



Note • The **Publication Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Publisher**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

Table 6-63 • Publication Manager Dialog Box Properties

Properties	Description
Name	Name of the Publication.
Description	Description of the Publication.
Status	The status of the last attempted update of the Publication: Succeeded or Failed. If the update is currently being executed, Executing appears in this column.
Last Action	Text describing the last action that was performed on this Publication.
Last Run	The date and time that the Publication was last updated.
Next Run	The next scheduled update time of that Publication.
New	Click to create a new Publication using the Publication Wizard.
Edit	Click to access the Publication Properties dialog box, where you can edit the selected Publication's name, description, scheduled update day and time, the type of data you want to include in the Publication, the list of users allowed to access the Publication, and the packages included in the Publication.
Run	Click to manually update the selected Publication. The Publication data is replicated from your Application Catalog to the Publisher Database Server, making it available for Subscription.
Delete	Click to delete the selected Publication from the Publication Manager.



Caution • If an attempt to create a Publication fails, it could be because your Publication server does not have a Distributor server configured. For more information, see [Initial Configuration Checklist](#).

Publication Processing Dialog Box

The Publication Processing dialog box opens after you click Finish on the **Publication Summary** panel of the Publication Wizard.

The Publication Processing dialog box tracks the progress of the Publication Wizard as the Publication is created, the snapshot agent is created, the permissions are added to the Publication Access List, and articles are added to the Publication.

When processing is finished, the new Publication is listed on the **Publication Manager** dialog box.

Publication Properties Dialog Box

The Publication Properties dialog box, which is accessed by clicking Edit on the **Publication Manager** dialog box, allows you to edit the selected Publication's name, description, scheduled update day and time, the type of data you want to include in the Publication, the list of users allowed to access the Publication, and the packages included in the Publication.

The Publication Properties dialog box contains the following tabs:

Table 6-64 • Publication Properties Dialog Box Tabs

Tab	Properties
Details Tab	<p>This tab contains the following properties:</p> <ul style="list-style-type: none">• Name—Name that identifies this Publication. This field is read-only.• Description—Enter a description of the purpose or intended subscribers to this Publication. This description will be listed on the Publication Manager.• Schedule—In this field, you specify how often you would like this Publication to be automatically updated. Click the Change button to access the Task Scheduling dialog box, where you can specify how frequently the Subscription will be automatically updated.
Options Tab	<p>Specify the following Publication data type options:</p> <ul style="list-style-type: none">• Groups and Package Data—Select this option if you want to select a specific set of packages from the various groups in the Application Catalog to include in the Publication. If you select this option, the Publication Packages panel will appear, prompting you to select packages to include in the Publication. If you do not select this option, all groups and all packages within those groups will be included in the Publication. This field is read-only.• Merge Module Data—Select this option if you would also like to include all Merge Modules associated with the packages you have selected in the Publication.
Access List Tab	<p>Specify the list of users allowed to access this Publication. It initially shows a list of users who have the necessary privileges to access the Publication: those users who have been assigned sysadmin privileges to this database server through the SQL Enterprise Manager. You can remove users from the list. See Publication Access List Panel for more information.</p>

Table 6-64 • Publication Properties Dialog Box Tabs (cont.)

Tab	Properties
Packages Tab	If Groups and Package Data is selected on the Options tab, this tab lists the Groups and Packages in the SQL database you are connected to. Select the packages that you want to include in this Publication. See Publication Packages Panel for more information.

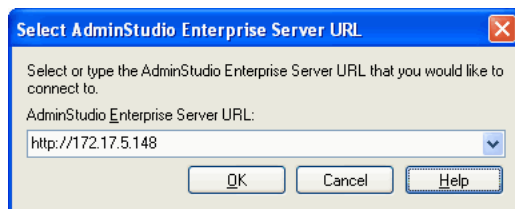
Select Application Catalog Dialog Box

The **Select Application Catalog** dialog box opens when you are attempting to connect to an existing Standalone Application Catalog that requires database authentication by selecting it from a list of recently used Application Catalogs.

- **Enterprise Server**—Select this tab to open the AdminStudio Enterprise Server Application Catalog database. See [Enterprise Server Tab](#).
- **Standalone**—Select this tab to open an Application Catalog other than the AdminStudio Enterprise Server database. See [Standalone Tab](#).
- **Recent**—Provides a list of recently opened Application Catalogs. When you select an Application Catalog and click **OK**, either the Application Catalog opens or you are prompted for login information (if you need authentication to the Application Catalog). See [Recent Tab](#).

Select AdminStudio Enterprise Server URL Dialog Box

If you click the HTTP link on an AdminStudio Enterprise Server Login dialog, this dialog box opens prompting you to identify the AdminStudio Enterprise Server URL that you would like to connect to.



Subscription Manager Dialog Box

The Subscription Manager lists all Publications that you are subscribed to in the SQL Server database you are currently connected to, and displays their Enable/Disable status.

Use this dialog box to view Subscription information, create a new Subscription, disable a Subscription so that it is not automatically updated, manually update a Subscription, and delete a Subscription.

While you can subscribe to multiple Subscriptions, only one Subscription can be active and listed in your Application Catalog at any one time: the one that has been executed (updated) the most recently. For information on how to use the Disable feature on the Subscription Manager to subscribe to more than one Application Catalog, see [Specialized User Scenarios](#).



Note • The **Subscription Manager** option under **Replication** on the AdminStudio **Catalog** menu is only enabled if the current user is connected to an SQL Server database, has been assigned sysadmin privileges in SQL Enterprise Manager on that database server, and the database was identified when it was created as being intended for use as a **Subscriber**. See [Application Catalog Wizard](#) for information on how to identify databases as Subscribers or Publishers.

The Subscription Manager dialog box contains the following properties:

Table 6-65 • Subscription Manager Dialog Box Properties

Properties	Description
Publisher	Name of the Publisher Database Server.
Publisher Database	Name of the database on the Publisher Database Server containing the Publication.
Publication	Name of the subscribed Publication.
Description	Description of the subscribed Publication that was entered by the creator of the Publication.
Status	The status of the last attempted update of the Publication: Succeeded or Failed. If the update is currently being executed, Executing appears in this column.
Last Action	Text describing the last action that was performed on this Subscription.
Last Run	The date and time that the Subscription was last updated.
Next Run	The next scheduled update time of that Subscription.
New	Click to create a new Subscription using the Subscription Wizard .
Enable/Disable	Click to toggle the selected Subscription's status between Enabled and Disabled. When you disable a Subscription, you are turning off its automatic update feature. Subscriptions that are enabled are preceded by a check mark, while disabled Subscriptions are preceded by a red X. For more information, see Specialized User Scenarios .
Run	Click to manually update the selected Subscription. The Subscription Manager will then replace the data of the Application Catalog's currently active Subscription with the most recent version of the selected Subscription's data. When a disabled Subscription is selected, the Run button is disabled. For more information, see Specialized User Scenarios .
Delete	Click to delete the selected Subscription from the Subscription Manager.



Caution • If you are unable to receive a Subscription, the most likely cause is that the Publication has not been published. Check to find out if the Publication has been published.

Task Scheduling Dialog Box

This dialog box is used to specify how frequently a Publication or Subscription will be automatically updated. Both the Publication Manager and Subscription Manager require schedule information. This dialog box is invoked by clicking the **Change** button on the **Details** panel of the **Publication** and **Subscription Wizards**, and the **Details** tab of the **Publication Properties** dialog box.



Note • A Publication's update frequency can be changed after the Publication is created by accessing the **Details** tab of the **Publication Properties** dialog box. However, a Subscription's update frequency is specified when the Subscription is created and cannot be changed.

Table 6-66 • Task Scheduling Dialog Box Properties

Properties	Description
Run the process every:	Select how frequently you want this Publication or Subscription to be automatically updated: Day, Week, or Month.
on	<p>This list offers different selections depending upon the choice you made in the Run the process every field:</p> <ul style="list-style-type: none"> Day—Only one selection is listed: Every Day. Week—Select the day of the week that you want to update this Publication or Subscription. Month—Select the day of the month that you want to update this Publication or Subscription.
At:	Enter the time of day that you want the update to occur.
Starting on:	<p>Select the day or date that you want this Publication to begin being automatically updated on the day or date you selected from the Run the process every list.</p> <p>For example, if you want the Publication to be automatically updated on the 15th of each month, you would select the first day of the first month that you want the automatic updates to begin, not the first day that the Publication should be updated. If it is Monday and you want the Publication to be updated every Friday, to have the Publication begin being updated on the upcoming Friday, you would select the current day of the week (Monday), but if you didn't want the Publication to be updated until the Friday after next, you would select Saturday.</p>

Click OK to exit the Task Scheduling dialog box.

Test Connection Dialog Box

When you are creating a new Application Catalog using the [Application Catalog Wizard](#), a **Test** button is provided on the [Specify Database Information Panel](#) to allow you to test the connection to the selected database server before completing the rest of the Wizard panels. When you click **Test**, this dialog box opens and informs you if the connection was successful.

Virtual Package Association Dialog Box

You can choose to use the **Associate Package** function in Application Manager to manually associate a virtual package with its source Windows Installer package after both packages have been imported into the Application Catalog.

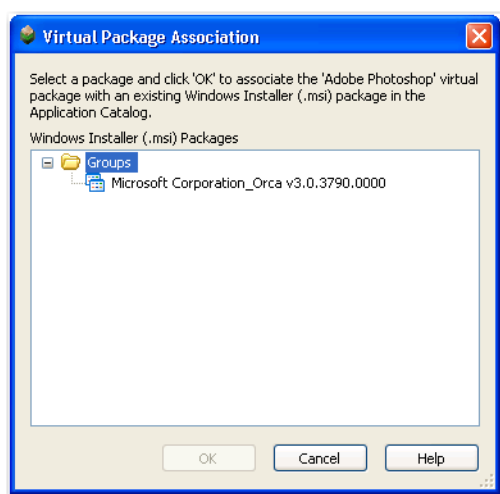


Figure 6-9: Virtual Package Association Dialog Box

If you select a virtual package in the Application Manager tree and then select **Associate Package** from the context menu, the **Virtual Package Association** dialog box opens, listing all of the Windows Installer packages in the Application Catalog. Select the virtual package's source Windows Installer package and click **OK**. The Windows Installer package will now be listed in the **Associated Packages** list in the Application Manager **Virtual Packages** view.



Important • After you have imported a virtual package into the Application Catalog, you can use the **Associate Package** function to associate it with any Windows Installer package in the Application Catalog, even one that is not its source package. Therefore, it is preferable to use the Import Wizard to import both the Windows Installer and virtual packages at the same time so that AdminStudio can create the proper associations.

Wizards

Application Manager includes the following Wizards:

- [Application Catalog Wizard](#)
- [Import Wizard](#)
- [Package Auto Import Wizard](#)
- [Merge Wizard](#)
- [Publication Wizard](#)
- [Subscription Wizard](#)
- [Upgrade Wizard](#)

Application Catalog Wizard

You use the Application Catalog Wizard to create a new SQL Server Application Catalog database. This Wizard includes the following panels:

- [Welcome Panel](#)
- [Specify Database Information Panel](#)
- [Select Software Repository Location Panel](#)
- [Creating Application Catalog Panel](#)

Welcome Panel

The first panel of the Application Catalog Wizard welcomes you to the Wizard.

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled.

Table 6-67 • Welcome Panel Options and Buttons

Button	Description
Include Sample Data in New Catalog	Select this option to automatically populate the new Application Catalog with sample package data. This allows you to immediately test AdminStudio features using the packages that have already been imported.
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.

Table 6-67 • Welcome Panel Options and Buttons

Button	Description
Help	Brings up help specific to the Import Wizard panel currently displayed.



Specify Database Information Panel

On the **Specify Database Information** panel of the [Application Catalog Wizard](#) and the **Standalone** tab of the [Connect Application Catalog Dialog Box](#), enter the information required to login to the specified Application Catalog.

Table 6-68 • Application Catalog Wizard / Specify Database Information Panel Options

Option	Description
Server	The list of available SQL Servers on the network. You can also manually enter the name of the SQL Server to which you want to connect.
Authentication	Select one of the following options: <ul style="list-style-type: none">• Windows Authentication—Choose to use Windows network authentication (your network login ID) to log into this Application Catalog.• Server Authentication—Choose to use SQL Server login identification for authentication.• Login ID and Password—If you chose Server Authentication, enter the appropriate Login ID and Password.
Catalog	Select the catalog from those available on the Server.
Test	Click this button to test whether a connection can be made to the database.

Table 6-68 • Application Catalog Wizard / Specify Database Information Panel Options (cont.)

Option	Description
Create database as a subscriber	<p>Select this check box to designate this new Application Catalog as a Subscriber database in Application Catalog Replication (a database that receives replicated data). When this option is not selected, or when an existing database is upgraded, the Application Catalog is defined as a Publisher database in Application Catalog Replication (a database that makes data available for replication).</p> <p>The AdminStudio interface selectively enables/disables the Replication menu items on the Catalog menu based upon the type of Application Catalog that is currently opened.</p>  <p>Edition • Application Catalog Replication is included with the AdminStudio Enterprise Edition. If you have AdminStudio Standard or Professional Editions, this check box is disabled. Contact your AdminStudio Sales Representative for more information.</p>  <p>Note • This check box only appears when creating a new Application Catalog, not when logging in to an existing Application Catalog.</p>

Select Software Repository Location Panel

A Windows Installer package is made up of many files that are executed when the setup is run. You only import the .msi file into the Application Catalog, not all of the files necessary for installation. With the Software Repository, when you import an installation package into the Application Catalog, all of the files associated with that package are copied into the Software Repository location, a directory that you specify. This allows you to manage those files, preventing them from getting modified or lost.

On the **Select Software Repository Location** panel, you can choose to **Enable the Software Repository** for the new Application Catalog, and specify a **Proxy Account** for AdminStudio to use to make modifications to the directory path selected as the **Software Repository Location**.

This panel includes the following options:

Table 6-69 • Select Software Repository Location Panel Options

Option	Description
Enable Software Repository	Select this option to enable the Software Repository feature for this new Application Catalog.
Software Repository Location	Specify the directory location of the Software Repository for this Application Catalog.

Table 6-69 • Select Software Repository Location Panel Options (cont.)

Option	Description
Proxy Account	Specify a Login ID and Password for AdminStudio to use when modifying the Software Repository Location directory.

Creating Application Catalog Panel

This panel displays the progress while your new Application Catalog is being created. If the Application Catalog cannot be created, an error message will be displayed.

Import Wizard

The Import Wizard allows you to import the following installation package types into the Application Catalog: Windows Installer packages and associated transform and patch files, virtual packages (Microsoft App-V, VMware ThinApp, and Citrix), merge modules, OS snapshots, Marimba Native Channel Packager files, and other non-MSI setup formats (such as InstallShield Professional or ISMP installations).

The Import Wizard consists of the following panels:

- [Welcome Panel](#)
- [File Selection Panel](#)
- [Associate Virtual to MSI Packages Panel](#)
- [MST Source Information Panel](#)
- [MSP Source Information Panel](#)
- [MSM Source Information Panel](#)
- [Virtual Package Import Panel](#)
- [Target Package Information Panel](#)
- [Target Package Information Panel \(Software Repository\)](#)
- [Target Package Information Panel \(Transforms/Patches\)](#)
- [Import Options Panel](#)
- [Additional non-MSI Import Options Panel](#)
- [Destination Group Panel](#)
- [Summary Panel](#)

When run, Application Manager displays its progress messages in the **Import** tab of the **Output Window**.



Note • Depending on configurations set in the ConflictSolver and Application Manager **Options** dialog box, validation and conflict identification may take place during import. See [Validating During Import](#) and [Checking for Conflicts During Import](#).

Welcome Panel

You can use the Import Wizard to import the following installation package types into the Application Catalog:

- [Windows Installer packages \(.msi\)](#)
- [Merge modules](#)
- [Marimba Native Channel Packager files \(.ncp\)](#)
- [Microsoft App-V applications \(.osd\)](#)
- [Citrix profiles \(.profile\)](#)
- [ThinApp applications \(.exe\)](#)
- [OS snapshots \(.osc\)](#)
- [Other setup types](#)

This panel, and others in the Wizard, have four buttons located at the bottom of the panel. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:

Table 6-70 • Welcome Panel Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help specific to the Import Wizard panel currently displayed.

File Selection Panel

On the **File Selection** panel, select the package file to import, and click **Next**. The following types of packages can be imported:

Table 6-71 • Import Package Types

Package Type	Description
Windows Installer Files (*.msi)	File that contains all of the information that the Windows Installer requires to install or uninstall an application or product and to run the setup user interface. The .msi file can also contain one or more transform files (.mst) and one or more patches (.msp).
Windows Installer Merge Module Files (*.msm)	Simplified Windows Installer .msi files. A merge module cannot be installed alone because it lacks some vital database tables that are present in an installation database. Merge modules also contain additional tables that are unique to themselves. To install the information delivered by a merge module with an application, the module must first be merged into the application's .msi file.
OS Snapshot Files (*.osc)	Files representing a particular computer system's contents. To generate an OS snapshot file, use the OS Snapshot Wizard to scan a computer's operating system and record the files, INI files, shortcuts, and registry entries present.
Native Channel Packager Files (*.ncp)	Files that are used to describe software components, their versions, their underlying structures, and their dependence on other components. Marimba's NCP file is derived from an OSD (Open Software Description) file and is currently used by Marimba's Castanet family of products.
Microsoft App-V (*.osd)	An application that runs within the Microsoft App-V virtual environment. When a Microsoft App-V application is created, the installation data from the original Windows Installer package is converted into an .sft file that is compatible with Microsoft App-V version 4.5. The installation shortcuts are converted into .osd files, which provide access to programs within the .sft file.
Citrix Profiles (*.profile)	An application that runs within a Citrix virtual environment, which prevents it from interfering with other software running on the same machine.
ThinApp Applications (.exe)	A virtual application that runs within a ThinApp virtual environment.
Other Non-MSI Setup Types	Non-MSI setup types (such as InstallShield Professional or ISMP installations).

Adding a File to the Software Repository

A Windows Installer package is made up of many files that are executed when the setup is run. However, only the .msi file is imported into the Application Catalog database. To safeguard these additional files against alteration or being misplaced, you can choose to manage these files using the Software Repository.

If you want to add the file to the Software Repository, select **Add the file(s) to the Software Repository** option on the **File Selection Panel**.



Note • The **Add the file(s) to the Software Repository** option is only displayed if you are connected to an Application Catalog that has the Software Repository enabled. See [Enabling the Software Repository in an Existing Application Catalog](#).

After you select this option, if AdminStudio determines that there are other files in that directory that could be associated with the selected file, the **Additional Dependencies of the Import File** list appears, enabling you to select any additional files to be managed by the Software Repository.

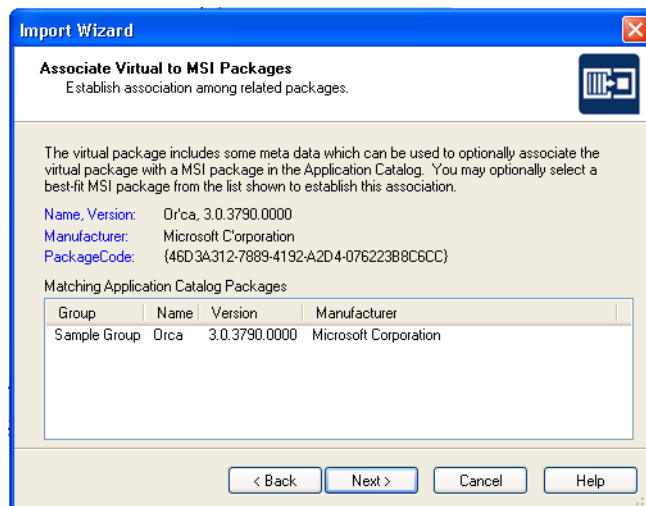
See [Using the Software Repository](#) for more information.

Associate Virtual to MSI Packages Panel

This panel opens if you are importing a virtual package and its source Windows Installer package exists in the Application Catalog. If this Windows Installer package has been imported into more than one location in the Application Catalog, multiple packages are listed.

Virtual packages are self-contained entities which ordinarily cannot be modified after they are created. By associating a virtual package with the Windows Installer package which originated it, you have the convenience of being able to easily locate the virtual package's originating Windows Installer package, modify the original Windows Installer package, and then regenerate the virtual package.

Select the package that you want to associate with the virtual package you are importing and click **Next** to continue.





Important • During virtual package conversion, AdminStudio 9.0 and later creates an XML file, *metadata.xml*, that contains metadata identifying the original Windows Installer package that was used to create the virtual package. However, if the virtual package was created by a method other than the AdminStudio 9.0+ conversion process, this metadata file will not have been created and this **Associate Virtual to MSI Packages** panel will not be displayed.

MST Source Information Panel


If you are importing a Windows Installer package, you can include additional transforms to be imported along with it.

When the **MST Source Information** panel opens, all of the .mst files that are in the same directory as the Windows Installer file you are importing are automatically listed in the **Transform Files (.mst)** list, but only those .mst files that AdminStudio determines are probably applicable to this Windows Installer package are selected to be included in the import. If you do not want to import a selected .mst file, clear the selection.

You can also add additional transform files:



Task: To add transform files:

1. Click the Browse () button in the Transforms area and select the transform file that you want to import.
2. If the package requires multiple transforms, you can repeat the procedure as necessary.
3. The order in which transforms are applied can be changed by selecting a transform and clicking the up and down arrows.
4. If you need to delete a transform you have added, clear the check box.
5. Click **Next** to continue.

MSP Source Information Panel

If you are importing a Windows Installer package, you can include additional patches to be imported along with it. If a patch file was already in the same directory as the Windows Installer file you are importing, that patch file will automatically appear in the **Patches (.msp or .exe)** list. If you do not want to import it, clear the selection.




Note • If you specify an *update.exe* patch file that was created by Developer/DevStudio/InstallShield Editor, Application Manager will extract the .msp file in the Temp folder and then perform the import.

You can also add additional patch files:



Task: *To add patch files:*

1. Click the Browse () button in the Patches area and select the patch file that you want to import.
2. If the package requires multiple patches, you can repeat the procedure as necessary.
3. The order in which patches are applied can be changed by selecting a patch and clicking the up and down arrows.
4. If you need to delete a patch you have added, clear the check boxes.
5. For patches to be applied to a Windows Installer package, it is necessary to perform an Administrative install of the Windows Installer package and then perform an Administrative install of each patch package one by one. This way, the content of each patch package is appended to the Windows Installer package at the Administrative install location.


Therefore, next to the **Specify the location for Administrative install** box, click the Browse button and specify a directory to provide Application Manager with the location where an Administrative Install of selected Windows Installer package will be performed.
6. Click **Next** to continue.

MSM Source Information Panel

If you are importing a Merge Module by selecting a Merge Module on the **Merge Modules** tab and selecting **Import Merge Module** from the context menu, this panel opens, allowing you to import multiple Merge Modules into the Application Catalog at one time.



Task: *To import merge modules from the Merge Modules tab:*

1. Open **Application Manager**.
2. Open the **Merge Modules** tab.
3. Select the root Merge Module or one of the imported Merge Modules and select **Import Merge Module** from the context menu. The **MSM Source Information** panel of the **Import Wizard** opens.
4. Click the Browse () button in the **Merge Modules** area and select the merge module file that you want to import.
5. To import multiple patches, you can repeat the procedure as necessary.
6. The order in which merge modules are applied can be changed by selecting a merge module in the list and clicking the Move Up and Move Down arrows.
7. If you need to delete a merge module you have added, clear its check box.
8. If you want to store this merge module in the Software Repository, select the **Add the file(s) to the Software Repository** option.

9. Click **Next**. The **Summary** panel opens.
10. Click **Finish** to accept these options and begin the import.

A report of the import process appears on the **Import** tab in the Output window.



Note • You can also import a Merge Module into the Application Catalog by selecting **Import Package** on the **Catalog** menu. However, using that method, only one Merge Module can be imported at a time.

Virtual Package Import Panel

This panel opens if the Import Wizard finds a virtual package in the same directory as the Windows Installer package you have selected to import.

The Import Wizard searches the selected package's folder and subfolders for virtual packages created by AdminStudio. If a virtual package is found that appears to have been created by AdminStudio, then it will be listed on this panel.

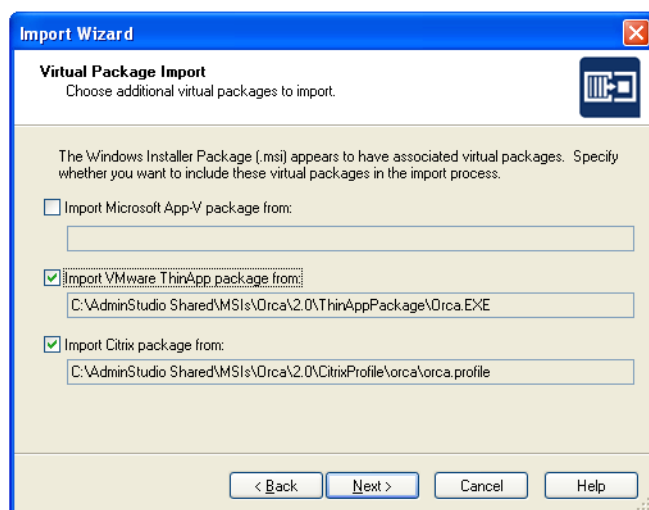


Figure 6-10: Import Wizard / Virtual Package Import Panel

Select the package or packages you want to import into the Application Catalog and click **Next** to continue.



Note • If a virtual package created by AdminStudio exists in a directory other than the one that contains the selected Windows Installer package, it will not be listed on this panel. After the import of the selected Windows Installer package is complete, you can use the Import Wizard to import virtual packages separately, as described in [Importing Virtual Packages](#). You would then be prompted to associate the virtual package with its already-imported source Windows Installer package.

Target Package Information Panel

This panel opens if you are attempting to import a package that has been identified as a duplicate to a package that is already in the Application Catalog (and is not stored in the Software Repository). The identification of a duplicate package is based upon the identifiers selected on the **Duplicate Package** tab of the ConflictSolver and Application Manager [Options Dialog Box](#).

An automatically generated new name is displayed in the **Package Name** field. You can either accept this generated name or enter a name of your choosing to identify the duplicate package before the package is imported.



Note • The name that is entered on the **Package Name** field is the name that will be displayed in the product tree on the left side of the ConflictSolver and Application Manager views. However, changing this “display” name does not change the Product Name that appears in the title bar of the ConflictSolver and Application Manager Product Views.

Target Package Information Panel (Software Repository)

This panel opens if you are attempting to import a package that has been identified as a duplicate to a package that is already in the Application Catalog (and is stored in the Software Repository). The identification of a duplicate package is based upon the identifiers selected on the **Duplicate Package** tab of the ConflictSolver and Application Manager [Options Dialog Box](#).

An automatically generated new name is displayed in the **Package Name** field. Because this package is part of the Software Repository, you have several options:

- The package should be treated as a new version of the existing package
- The package should be treated as a new package named [the name entered in the **Package Name** field]
- The package should overwrite the existing package version

Make your selection and click **Next**.

For more information, see [Using the Software Repository](#).



Note • The name that is entered on the **Package Name** field is the name that will be displayed in the product tree on the left side of the ConflictSolver and Application Manager views. However, changing this “display” name does not change the Product Name that appears in the title bar of the ConflictSolver and Application Manager Product Views.

Target Package Information Panel (Transforms/Patches)

This panel opens if you are attempting to import a transform or patch file that has been identified as a duplicate to a file that is already in the Application Catalog (and is stored in the Software Repository).

- **Updated file**—If this file is an updated version of a transform or patch file that has already been imported into the Application Catalog, select the **Reimport new version of the package with this file change** option.
- **File associated with a different package**—If this file has the same name as a transform or patch file that has already been imported into the Application Catalog, but is not associated with the same Windows Installer package as the existing file, select the group that contains the Windows Installer package that this file is associated with.

Import Options Panel

The **Import Options** panel allows you to select custom tables that are not part of a standard Windows Installer package, and direct Application Manager to import them. Indicate which custom tables you want to import by selecting the check box next to the table name. You can also use the Select All and Clear All buttons to make selections.

To add one of the listed tables to the **Ignore Tables** list on the **Import** tab of the ConflictSolver and Application Manager [Options Dialog Box](#) (so that it will not be imported into the Application Catalog during any import), select the table name and click **Add to Ignore List**.

Additional non-MSI Import Options Panel

If you are importing a non-Windows Installer package, it is necessary to enter a name for the package and the base directory for the package (where all of its installation files are located).

Table 6-72 • Import Wizard / Additional non-MSI Import Options

Options	Description
Product Name	Enter a name to identify this package in AdminStudio.
Directory	Click the Explore button to select the base directory of this package where all of its associated files can be found.
Include entire directory hierarchy	Select this option to include the selected directory and all of its subdirectories.

Destination Group Panel

The Destination Group panel allows you to specify the group (or groups) into which you want to import the MSI package.

You also have the option of selecting all groups by clicking the Select All button, or deselecting groups by clicking the Clear All button. If a group you want to import the package into does not exist, click the New button to create and configure it in the Group Properties dialog box.



Note • This panel is only displayed when launching the Import Wizard from the Catalog menu. If you right-click on a group in the Application Manager Product View and import a package or snapshot, the destination group will be group on which you right-clicked.

Summary Panel

Before executing the import, review the information in the Summary panel about the options selected in the previous panels.

Depending on the import type and how the Import Wizard was invoked, clicking Back returns you to the **Destination Group** panel, **MSM Source Information** panel, **OS Snapshot Information** panel, or **Other Setup Installation Files** panel. Click **Finish** to begin import.

Package Auto Import Wizard



Note • To use the Package Auto Import feature, you must have AdminStudio 9.5 Administrator permission. For all other users, the Package Auto Import options will be disabled.

The Package Auto Import feature allows you to copy packages between various Application Catalogs and maintain consistency between the copied packages. This allows you to maintain multiple Application Catalogs that match your corporate organization and optimize the performance of AdminStudio.

The Package Auto Import Wizard consists of the following panels:

- Welcome Panel
- Package Auto Import Type Selection Panel
- Remote Link Application Catalog Panel
- Folder Panel
- Groups Panel
- Schedule Panel
- Summary Panel

When run, Application Manager displays the output report in the Package Auto Import tab of the Output Window.



Tip • When you link to a package in a Remote Application Catalog, Application Manager connects to the Remote Application Catalog and imports an MSI file into the local Application Catalog. In order for Package Auto Import to work properly, both the Remote Application Catalog and the Remote MSI file must be accessible to the linked users at all times. In other words, both the Remote Application Catalog and all of the MSI files that have been imported into it must be able to be accessed by linked users via a shared network drive.

Welcome Panel

The first panel of the Package Auto Import Wizard welcomes you to the Wizard.

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:

Table 6-73 • Welcome Panel Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific Package Auto Import Wizard panel.

Package Auto Import Type Selection Panel

On this panel, select the type of monitoring process that you want to configure. Select one of the following options:

- **Remote Application Catalog**—Monitor Windows Installer packages in a Remote Application Catalog and automatically import/re-import those packages at scheduled intervals. Application Manager can link to one or multiple Application Catalogs, and can link to one or multiple packages within those Application Catalogs.
- **Network Directory**—Automatically import or re-import all Windows Installer packages in a monitored network directory into your Application Catalog at scheduled intervals.

Remote Link Application Catalog Panel

The Remote Link Application Catalog panel of the Package Auto Import Wizard lists Application Catalogs that you have recently accessed.

To select the Application Catalog that you want to link to, you can perform either of these options:

- Select the Application Catalog you want to link to from the **Catalogs** list.
- Click **Browse** to open the [Select Application Catalog Dialog Box](#) and connect to an existing Application Catalog as described in [Connecting to an Existing Application Catalog](#).

During the linking process of an SQL Server Application Catalog, Application Manager will save the name of the database server, authentication mode, and name of the Application Catalog. Passwords are not saved.

Folder Panel

On the **Folder** panel, use the **Browse** button to select the network directory folder that you want to monitor.

To also monitor the selected folder's subfolders, select the Also include sub-directories option.



Note • While you are permitted to enter a local directory (rather than a network directory) as the monitored directory, this might cause problems if a person monitoring this directory has the same directory on his local machine. For example, if a directory named C:\MyAppCatalogs is chosen to be the monitored directory, and a person who choose to monitor this directory also has a directory on his local machine named C:\MyAppCatalogs, Application Manager would monitor that person's local directory rather than the directory that was chosen to be the monitored directory.



Note • If you select this option, when packages in subdirectories are imported into your Application Catalog, they will be placed in groups that maintain the folder hierarchy. For example, if you are monitoring a directory named Marketing, and that directory has a subdirectory named Graphics that contains packages, when those packages are imported into your Application Catalog, they will be placed in the Graphics subgroup of the destination group.

Groups Panel

The **Groups** panel of the Package Auto Import Wizard lists the group structure and the packages that are in the specified Remote Application Catalog. Choose the packages that you want to link to by selecting the check box next to the package name.

Click **Next** to continue.

Destination Group Panel

When you are setting up Network Directory monitoring, the **Destination Group** panel opens, prompting you to select the Application Catalog group into which the packages will be imported.

Select the group in the Application Catalog where you want these packages to be imported, and click Next.



Note • Packages that are imported using Remote Application Catalog monitoring are imported into the Package Auto Import group in the Application Catalog.

Schedule Panel

On the **Schedule** panel of the Package Auto Import Wizard you can choose to enable automatic import of linked packages (or packages in a monitored Network Directory), and can also specify how often the Remote Application Catalog or Network Directory will be checked for updates.

The following options are included:

Table 6-74 • Schedule Panel Properties

Option	Description
Schedule AdminStudio to update the changed packages	Select this option to enable automatic import of the linked packages (or packages in the monitored Network Directory) at scheduled intervals. When this option is selected, the rest of the scheduling fields on this panel are enabled.
Run the process	Select Daily or Weekly to specify how often you want the Remote Application Catalog or Network Directory to be checked for updates.
At	Select the time of day that you want the automatic import to occur by selecting the hour, minutes, seconds, or AM/PM setting and then clicking the up and down arrows to adjust the setting.
Starting on	Click on the arrow to open the Calendar and then select the first day that you want automatic importing to occur.

Summary Panel

The **Summary** panel provides a detailed summary of the options that were selected in the previous panels of the Wizard. Click **Finish** to complete the remote link. Clicking **Back** returns you to the previous panel.

Merge Wizard

The Merge Wizard, available by clicking **Merge** on the **Catalog** menu, allows you to merge application data between two Application Catalogs.

The source Application Catalog's data is merged into the currently open Application Catalog. This is a one-time-only database merge. The merged data is not linked in any way to the source Application Catalog.

Merging could be used in an organization where multiple people test packages. After the packages have been tested, all of the “tested” applications could be consolidated into one centralized Application Catalog. Another reason to merge Application Catalogs would be a need to have the same data available in multiple geographic locations.

The following panels are included in the Merge Wizard:

- [Welcome Panel](#)
- [Source Application Catalog Panel](#)
- [Groups Panel](#)
- [Progress Panel](#)

Welcome Panel

The Merge Wizard allows you to merge application data between two Application Catalogs. The source Application Catalog's data is merged into the currently open Application Catalog.

The first panel in the Merge Wizard is the **Welcome** panel. Click **Next** to proceed to the **Source Application Catalog** panel.

Source Application Catalog Panel

The **Source Application Catalog** panel lists all of the Application Catalogs that the user has permission to view. Select the Application Catalog that contains the data that you want to merge.

If the Application Catalog that you select requires additional authentication, the [Select Application Catalog Dialog Box](#) opens. Connect to that catalog as described in [Connecting to an Existing Application Catalog](#).

Groups Panel

From the **Groups** panel, you can select the groups you want to merge. The available groups displayed are from the source Application Catalog, and data from the selected groups will be merged into the current Application Catalog. If you select the option to **Replace duplicate application data**, if a duplicate package exists in the current Application Catalog, the source Application Catalog's package will overwrite the duplicate.

Progress Panel

The **Progress** panel displays messages from the merge process.

When the merge is complete, review the log as necessary and click **Finish** to exit the Wizard.

OS Snapshot Wizard

The following topics contain information about each Wizard panel and dialog box available through the OS Snapshot Wizard. The help topics in this reference are the same detailed documentation that is displayed when you press the F1 key or click the Help button while working in a Wizard or dialog box.

Select one of the following links for OS Snapshot reference information:

- [Welcome Panel](#)
- [Project Information Panel](#)
- [Analyzing Panel](#)
- [OS Snapshot Summary Panel](#)
- [Analysis Options Dialog Box](#)
- [ISSnapshot.ini File](#)

Welcome Panel

The **Welcome** panel appears when you first launch the OS Snapshot Wizard, providing some introductory information about the use of the OS Snapshot Wizard.

The **Next** button advances you to the **Project Information** panel.

Project Information Panel

The Project Information panel gathers information necessary for taking the OS Snapshot.

You must provide the following information before the Start button is enabled, allowing you to take the snapshot.

Table 6-75 • Project Information Panel Options

Option	Description
OS Snapshot project name	Provide a name for the snapshot file (.osc).
OS Snapshot project folder	Provide the directory in which snapshot data will be stored. Either enter the path in the field, or click the Browse (...) button to navigate to it. If the directory already exists, a confirmation dialog box opens when you click the Next button.

If you want to review or change current capture settings, click Edit to display the Analysis Options dialog box.

Analyzing Panel

The **Analyzing** panel appears while the OS Snapshot Wizard analyzes your system.

Following the snapshot, the **Summary** panel appears.

OS Snapshot Summary Panel

At the end of the OS Snapshot process, the Summary panel is displayed, containing information about the OS Snapshot that was just performed.

Prior to clicking Finish, review the information to ensure the snapshot contains the data you expected.

Following the OS Snapshot process, you can import the snapshot into the Application Catalog and use it as a baseline to which setups can be compared.



Caution • OS Snapshots should only be used for comparison in ConflictSolver. You should never attempt to convert an OS Snapshot into an MSI package.

Analysis Options Dialog Box

The **Analysis Options** dialog box, accessible by clicking **Edit** from the **Project Information** panel, allows you to specify capture types for the OS snapshots.

You can select the following:

- Files
- INI files
- Shortcuts
- Registry data

Additionally, you can restrict directory analysis to specific directories, which can significantly improve OS Snapshot Wizard performance. Click New to add a directory restriction, edit to modify an existing restriction, or delete to remove a restriction.

Options set in this dialog box apply to the current and subsequent snapshot sessions.

ISSnapshot.ini File

The ISSnapshot.ini file is the default exclusion file for the OS Snapshot Wizard. It contains exclusions to be applied when capturing an OS snapshot, and mainly focuses on specific items that should not be included in applications, such as InstallShield Professional-specific COM settings and OS Snapshot-specific registry entries.

The file is located in the Windows folder, and can be edited using the Exclusions Editor, or using a text editor. See [Exclusions Editor Interface](#).



Note • It is strongly recommended that you not modify this file, as it increases the likelihood of either inadvertently omitting necessary pieces of the OS snapshot, or including registry entries or files that should not be part of the snapshot.

Publication Wizard

The Publication Wizard for Application Catalog Replication walks you through the process of creating an SQL Server Publication from an Application Catalog database.

The Publication Wizard consists of the following panels:

- [Welcome Panel](#)
- [Publication Details Panel](#)
- [Publication Data Options Panel](#)
- [Publication Access List Panel](#)
- [Publication Packages Panel](#)
- [Publication Summary Panel](#)

Welcome Panel

The first panel of the Publication Wizard welcomes you to the Wizard.

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:


Table 6-76 • Welcome Panel Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific Wizard panel.

Publication Details Panel

On the **Publication Details** panel, you are prompted to enter basic Publication properties of name, description, and the schedule. The **Publication Details** panel contains the following fields:

Table 6-77 • Publication Wizard / Publication Details Panel Options

Option	Description
Name	<p>Enter a name to identify this Publication. If you enter a duplicate name of an existing Publication, you will receive an error message.</p>  <p>Caution • The Publication Name cannot be a T-SQL reserved word.</p>
Description	<p>Enter a description of the purpose or intended subscribers to this Publication. This description will be listed on the Publication Manager.</p>
Schedule	<p>In this field, you specify how often you would like this Publication to be automatically updated. When a Publication is updated, all of its data is replaced by replicated data from the Publisher's local Application Catalog database.</p> <p>This field is not directly editable. Click the Change button to access the Task Scheduling dialog box, where you can specify how frequently a Publication will be automatically updated.</p>

Publication Data Options Panel

The **Publication Data Options** panel prompts you to specify the particular types of data to include in the Publication. This panel contains the following fields:

Table 6-78 • Publication Wizard / Publication Data Options Panel Properties

Properties	Description
Groups and Package Data	<p>Select this option if you want to select a specific set of packages from the various groups in the Application Catalog to include in the Publication. If you select this option, the Publication Packages panel opens, prompting you to select packages to include in the Publication. If you do not select this option, all groups and all packages within those groups will be included in the Publication.</p>
Merge Module Data	<p>Select this option if you would also like to include all Merge Modules in the Application Catalog in the Publication.</p>

Publication Access List Panel

On the **Publication Access List** panel, you specify the list of users allowed to access this Publication. It initially shows a list of users who have the necessary privileges to access the Publication: those users who have been assigned sysadmin privileges to this database server through the SQL Enterprise Manager. You can remove users from the list.

The **Publication Access List** panel contains the following options:

Table 6-79 • Publication Wizard / Publication Access List Panel Options

Option	Description
Type	Displays an icon to identify each listing as either a user or a group.
Name	Lists the name of the user or group.
Remove	Click to remove a user or group from the list. If a user is removed from the list, they will no longer be able to access the Publication, but they still would be able to be a user of the SQL Server and Publication database.



Caution • Only users that have been assigned sysadmin privileges in SQL Enterprise Manager to the connected database server can establish the Publisher-Subscriber relationship between the databases and are therefore listed on this panel. You can remove a user from this list (so that they will not be able to access this Publication), but you cannot add additional users to this list using AdminStudio. To add new users, you must use SQL Enterprise Manager.

Click Next to access the **Publication Packages** panel (if you selected the **Groups and Package Data** option on the **Publication Data Options** panel) or the **Publication Summary** panel if you did not select that option.

Publication Packages Panel

On the **Publication Packages** panel, the Groups and Packages in the Application Catalog you are connected to are listed. Select the packages that you want to include in this Publication.



Caution • You do not have to include all of the packages in the Application Catalog in this Publication. However, when this Publication is replicated in the Subscriber's Application Catalog, all of the Groups in the Publisher's Application Catalog will be listed, even if they do not contain any packages.



Caution • When a user subscribes to this Publication, they are subscribing to all of the packages in all of the Groups that the Publisher has included. However, if the Publisher adds a new package to one of these same Groups after he has created the Publication, the new package is not automatically added to the existing Publication, making

it unavailable for Subscription. To publish this new package, a new Publication would have to be created or an existing Publication would have to be edited.



Note • This panel only appears if you selected the **Groups and Package Data** option on the **Publication Data Options** panel.

Publication Summary Panel

This panel provides a detailed summary of the options that were selected in the previous panels of the Wizard

The **Publication Summary** panel contains the following options:

Table 6-80 • Publication Wizard / Publication Summary Panel Options

Option	Description
Generate publication as script	To generate an SQL script of the code necessary to create this Publication rather than actually create it, select this option and enter a path name for the script file. For more information, see Replicating Application Catalogs in Controlled SQL Database Environments .
Publish the publication immediately after creation	To publish this Publication immediately after the Publication Wizard is finished, select this option. The name of this Publication would be listed on the Publication Manager with a status of Executing, and the Publication data would be replicated from your Application Catalog to the Publisher Database Server—making it available for Subscription immediately. If you do not select this option, you would have to click Run on the Publication Manager to publish this Publication manually, or the Publication would automatically be published on its scheduled date.

Click Finish to complete the creation of this Publication.

Subscription Wizard

The Subscription Wizard for Application Catalog Replication walks you through the process of creating an SQL Server Subscription to an Application Catalog database.

The Subscription Wizard consists of the following panels:

- [Welcome Panel](#)
- [Subscription Details Panel](#)
- [Subscription Selection Panel](#)
- [Subscription Summary Panel](#)

Welcome Panel

The first panel of the Subscription Wizard welcomes you to the Wizard. This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:


Table 6-81 • Welcome Panel Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific Wizard panel.

Subscription Details Panel

On the **Subscription Details** panel, you are prompted to enter connection information for the Publisher database and scheduling information. This panel contains the following fields:

Table 6-82 • Subscription Details Panel Properties

Properties	Description
Database	<p>From the Catalogs list, select the Publisher Application Catalog you want to subscribe to, or click Browse to open the Select Application Catalog Dialog Box and connect to an existing Publisher Application Catalog as described in Connecting to an Existing Application Catalog</p>  <hr/> <p>Note • You must connect to a SQL Server database that was identified as being intended for use as a Publisher when it was created. See Application Catalog Wizard.</p>
Schedule	<p>In this field, you specify how often you would like this Subscription to be automatically updated. When a Subscription is updated, all of its data is replaced by replicated data from the Publisher database.</p> <p>This field is not directly editable. Click the Change button to access the Task Scheduling dialog box, where you can specify how frequently the Subscription will be automatically updated.</p>

Subscription Selection Panel

The **Subscription Selection** panel displays the Publications available from the selected Publisher database that satisfy the **Publication Access List** requirements for the currently connected user. In other words, it lists the Publications that this user has permission to subscribe to.

Select one Publication from this list to subscribe to and click **Next** to continue to the **Summary** panel.

Subscription Summary Panel

This panel provides a detailed summary of the options that were selected in the previous panels of the Wizard.

To generate an SQL script of the code necessary to create this Subscription rather than actually create it, select the Generate subscription as script option and enter a path name for the script file. For more information, see [Replicating Application Catalogs in Controlled SQL Database Environments](#).

Click **Finish** to complete the creation of this Subscription.

Upgrade Wizard

When you attempt to open an AdminStudio 5.0, 5.5, 6.0, 7.0, or 7.5 Application Catalog in AdminStudio 9.5, you are prompted to upgrade it to use the AdminStudio 9.5 schema.

Log files for the upgrade are created in the following directory:

AdminStudio Shared Directory\ConflictSolver\Logs



Note • Note the following regarding upgrading an existing Application Catalog:

- The upgrade of AdminStudio 3.0, 3.01, and 3.5 databases is not supported by AdminStudio 7.0 or later.
- Starting with AdminStudio 8.0, Microsoft Access databases are not supported.
- Starting with AdminStudio 9.01, Oracle databases are not supported.
- When an SQL Server Application Catalog database is upgraded, the old tables are not dropped from the Application Catalog.

The Upgrade Wizard consists of the following panels:

- **Welcome Panel**—Initial panel displayed when the Upgrade Wizard is launched. Click **Next** to proceed with the upgrade.
- **Progress Panel**—Displays the results of the upgrade. Click **Finish** to exit the Upgrade Wizard.

Upgrading Pre-AdminStudio 5.0 Application Catalogs

Pre-AdminStudio 5.0 Application Catalogs cannot be upgraded automatically by AdminStudio 7.0 or later. However, you can upgrade them using the Legacy Upgrade Wizard, a standalone utility that was included with AdminStudio 7.0 and 7.5. The Legacy Upgrade Wizard utility is installed in the following directory:

C:\Program Files\InstallShield\AdminStudio\7.x\Common\LegacyUpgradeWizard.exe

If you do not have a copy of AdminStudio 7.0 or 7.5 available to you, contact Technical Support.

AdminStudio Database Schema Tables

The AdminStudio Database Schema stores all data used by the Application Catalog. The data is stored in tables based upon how it is used. The [Table Prefixes](#) identify the purpose of the table. For example, AdminStudio System Table names begin with cssys. Each table contains one [Primary Key](#) and can contain one or many [Foreign Keys](#).

The section contains information on the following:

- [Database Server Permissions](#)
- [AdminStudio MSI Table Mapping](#)
- [Application Catalog Replication Tables](#)
- [cssys: AdminStudio System Tables](#)
- [cstbl: AdminStudio Supporting Tables](#)
- [as: Job Manager Tables](#)
- [cspch: OS Security Patch Wizard Tables](#)
- [pdt: Predeployment Test Tables](#)
- [Miscellaneous Tables](#)

Table Prefixes

The following table lists the prefixes used to name tables in the AdminStudio Database Schema, and describes the purpose of each group of tables:

Table 6-83 • AdminStudio Database Schema Table Prefixes

Prefix	Meaning
assys	These tables are AdminStudio system tables (such as tables used for security, roles, etc.)
csmsi	When Application Manager imports a MSI package, it will import all entries from all MSI tables within the MSI package. This data will be stored in the Application Catalog database with the same MSI table name prefixed by “csmsi”. The columns within the Application Catalog table exactly match the columns in the original MSI table. However, there are two additional columns added to the Application Catalog tables: a RowID column for use as a Primary Key, and a PkgRowID_ column for use as a Foreign Key into the cstblPackage table.
csmsm	When Application Manager imports a Merge Module (MSM file), it will import all entries from all Merge Module tables within the MSM file. This data will be stored in the Application Catalog database with the same MSM table name prefixed by “csmsm”. The columns within the Application Catalog table exactly match the columns in the original MSM table. However, there are two additional columns added to the Application Catalog tables: a RowID column for use as a Primary Key, and a ModuleRowID_ column for use as a Foreign Key into the csmsmModuleSignature table.
csosc	These are tables in the Application Catalog database that contain OS snapshot imported data. These four csosc tables—csoscFile, csoscIniFile, csoscRegistry and csoscShortcuts—are equivalent to the MSI tables of the same name, and contain data that is a conceptual representation of the four types of data in an operating system. The columns within the Application Catalog tables exactly match the columns in the original MSI tables. However, there are two additional columns added to the Application Catalog tables: a RowID column for use as a Primary Key, and a PkgRowID_ column for use as a Foreign Key into the cstblPackage table.
cssys	These are Application Catalog system tables (tables used to store schema information, extended attributes schema information, etc.). For a detailed list of Application Catalog system tables, see cssys: AdminStudio System Tables .
cstbl	These are Application Catalog supporting tables (tables used to store package, groups, conflict persistence, etc.). For a detailed list of supporting tables, see cstbl: AdminStudio Supporting Tables .
wftbl	These are AdminStudio workflow tables (tables used to store workflow, steps, projects, tools, etc.)
ams_	AdminStudio Workflow Manager stores the data in tables whose names begin with ams_.

Table 6-83 • AdminStudio Database Schema Table Prefixes

Prefix	Meaning
cssp_	These are the tables that contain the stored procedures used to access data from Application Catalog tables. The ASDataService Web service uses these stored procedures.
wfsp_	These are the tables that contain the stored procedures used to access data from Workflow tables. The ASDataService Web service uses these stored procedures.
amssp_	These are the tables that contain the stored procedures used to access data from Workflow Manager tables. The AMSDataService Web service uses these stored procedures.
as	Job Manager stores data in tables whose names begin with as .
cspch	OS Security Patch Wizard stores data in tables whose names begin with cspch .
pdt	Predeployment Test stores data in tables whose names begin with pdt .

Terminology

The following terms are used in this section to describe table columns:

Primary Key

A Primary Key is a column whose values uniquely identify each row in a table. A table can have only one Primary Key, and this column must contain a value. In the Application Manager Database Schema, the Primary Key is always the RowID column.

Foreign Key

A Foreign Key is a column or combination of columns used to establish and enforce a link between two tables. In the Application Manager database, the **PkgRowID_** column is the Foreign Key for MSI packages into the **cstblPackage** table—the table that is the primary table containing product information.

Database Server Permissions

In order to operate some AdminStudio tools, AdminStudio users require specific database permissions. Depending upon the type of user, you may wish to be more selective in the permissions you assign to these users.

If you have AdminStudio Enterprise Edition, you can assign permissions to individual users using the Role functionality in AdminStudio Enterprise Server, as described in [AdminStudio and Workflow Manager Roles and Permissions](#). Otherwise, you can provide more selective restrictions at the database server level using the information in the following table, AdminStudio Database Server Permissions.

Every AdminStudio user will need at a minimum read privilege to every table in the Application Catalog. The minimum permissions are described below, based upon the type of operation you want the user to perform.

Table 6-84 • AdminStudio Database Server Permissions

Type	Description
General User Administrative Process	<p>General administrative processes cover a range of activities such as adding groups, moving packages around, adding comments, updating extending attributes, etc. For example, these tables include cstblPackage, cstblGroups, and cstblGroupPackages. Any Application Catalog table which is not referenced explicitly in the discussion for the other AdminStudio processes should be considered a general user administrative table.</p> <p>Most AdminStudio users should have write access to these tables.</p>
Import Process	<p>The user importing MSI packages, merge modules, or just about anything will require write access to a significant set of Application Catalog tables depending on the type of import. For example:</p> <ul style="list-style-type: none">• MSI package file—For MSI package file import operations, those Application Catalog tables with a csmsi prefix are populated.• Merge modules—For merge module import operations, the csmsm prefixed tables are used.• Patches—For patch import operations, the cspch prefix tables are used.• OS snapshots—For OS snapshot import operations, the osc prefix tables are used. <p>Occasionally, the Import process will need to create new tables in the Application Catalog to support the import of custom table data. Either the necessary server privileges need to be given for these users, or the import of custom table data needs to be disabled within the Application Manager Options dialog box.</p>
Validation Process	<p>For this process, the user will need to be able to write entries into the cstblValidationResults and cstblValidationConfiguration tables.</p>
Dependency Scanning Process	<p>For this process, the user will need to be able to write entries into the cstblPackageExeDependencies table.</p>

Table 6-84 • AdminStudio Database Server Permissions (cont.)

Type	Description
Conflict Detection and Resolution Process	For this process, the user will need to be able to write entries into the cstblConflict prefixed table names.
Patch Impact Analysis Process	For this process, the user will need to be able to write entries into the cstblPatchConflict prefixed table names. This process will create and delete some temporary tables and, as such, the user performing this process should have the necessary server privileges to perform these operations.
Package Auto Import Process	The Package Auto Import process will ultimately generate a series of Import operations, and so the user performing these operations should have the Import process rights described above. If the user wants to edit these operations in the Wizard, then they will need write accession to the cstblSubscribed prefixed tables.
Workflow Operations Process	For this process, the user will need to be able to write entries into the wftbl prefixed table names.
Tools Properties Operations	For this process, the user will need to be able to write entries into the wftblTools table name.
Job Manager Process	For this process, the user will need to be able to write entries into the As prefixed table names.
Pre-Deployment Testing	For this process, the user will need to be able to write entries into the pdt prefixed table names.



Note • A number of processes within AdminStudio generate data which can subsequently be deleted by the AdminStudio user. Any discussion of the minimum privileges required for a specific AdminStudio process will also imply the privileges to delete this same data.

AdminStudio MSI Table Mapping

The AdminStudio Application Catalog is primarily organized using variations on Windows Installer table names. These are prefixed with “csmsi” for Application Manager MSI tables and “csmsm” for Application Manager merge module table. For example, the MSI table “Shortcut” is represented as “csmsiShortcut” in the Application Catalog.

Following is a complete listing of tables in an Application Catalog:

- [System Tables](#)
- [ConflictSolver Tables](#)
- [MSI Tables](#)
- [MSM Tables](#)
- [OS Snapshot Tables](#)
- [Workflow Tables](#)
- [Security Tables](#)
- [Workflow Manager Tables](#)

System Tables

An Application Catalog contains the following System Tables:

Table 6-85 • System Tables

Table	Corresponding Windows Installer Table
cssysExtendedAttributeSchema	N/A
cssysSchemaInfo	N/A
cssysTables	N/A

ConflictSolver Tables

An Application Catalog contains the following ConflictSolver Tables:

Table 6-86 • ConflictSolver Tables

Table	Corresponding Windows Installer Table
cstblPackage	N/A
cstblGroups	N/A
cstblGroupPackages	N/A

Table 6-86 • ConflictSolver Tables (cont.)

Table	Corresponding Windows Installer Table
cstblConflictConfiguration	N/A
cstblConflictACE	N/A
cstblConflictPackages	N/A
cstblConflictResults	N/A
cstblConflictTransformList	N/A
cstblConflictResultsDestPkgs	N/A
cstblExtendedAttributeData	N/A
cstblIgnoreTableList	N/A
cstblPackageHistoryLog	N/A
cstblTransformFile	N/A

MSI Tables

An Application Catalog contains the following MSI Tables:

Table 6-87 • MSI Tables

Table Name	Corresponding Windows Installer Table
csmsiActionText	ActionText
csmsiAdminExecuteSequence	AdminExecuteSequence
csmsiAdminUISequence	AdminUISequence
csmsiAdvtExecuteSequence	AdvtExecuteSequence
csmsiAdvtUISequence	AdvtUISequence
csmsiAppId	AppId
csmsiAppSearch	AppSearch
csmsiBBControl	BBControl
csmsiBillboard	Billboard

Table 6-87 • MSI Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsiBinary	Binary
csmsiBindImage	BindImage
csmsiCCPSearch	CCPSearch
csmsiCheckBox	CheckBox
csmsiClass	Class
csmsiComboBox	ComboBox
csmsiCompLocator	CompLocator
csmsiComplus	Complus
csmsiComponent	Component
csmsiCondition	Condition
csmsiControl	Control
csmsiControlCondition	ControlCondition
csmsiControlEvents	ControlEvents
csmsiCreateFolder	CreateFolder
csmsiCustomAction	CustomAction
csmsiDialog	Dialog
csmsiDirectory	Directory
csmsiDrLocator	DrLocator
csmsiDuplicateFile	DuplicateFile
csmsiEnvironment	Environment
csmsiError	Error
csmsiEventMapping	EventMapping
csmsiExtension	Extension

Table 6-87 • MSI Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsiFeature	Feature
csmsiFeatureComponents	FeatureComponents
csmsiFile	File
csmsiFileSFPCatalog	FileSFPCatalog
csmsiFont	Font
csmsiIcon	Icon
csmsiIniFile	IniFile
csmsiIniLocator	IniLocator
csmsiInstallExecuteSequence	InstallExecuteSequence
csmsiInstallUISequence	InstallUISequence
csmsiIsolatedComponent	IsolatedComponent
csmsiLaunchCondition	LaunchCondition
csmsiListBox	ListBox
csmsiListView	ListView
csmsiLockPermissions	LockPermissions
csmsiMedia	Media
csmsiMIME	MIME
csmsiModuleComponents	ModuleComponents
csmsiModuleDependency	ModuleDependency
csmsiModuleExclusion	ModuleExclusion
csmsiModuleSignature	ModuleSignature
csmsiMoveFile	MoveFile
csmsiMsiAssembly	MsiAssembly

Table 6-87 • MSI Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsiMsiAssemblyName	MsiAssemblyName
csmsiMsiDigitalCertificate	MsiDigitalCertificate
csmsiMsiDigitalSignature	MsiDigitalSignature
csmsiMsiFileHash	MsiFileHash
csmsiMsiPatchHeaders	MsiPatchHeaders
csmsiODBCAttribute	ODBCAttribute
csmsiODBCDataSource	ODBCDataSource
csmsiODBCDriver	ODBCDriver
csmsiODBCSourceAttribute	ODBCSourceAttribute
csmsiODBCTranslator	ODBCTranslator
csmsiPatch	Patch
csmsiPatchPackage	PatchPackage
csmsiProgId	ProgId
csmsiProperty	Property
csmsiPublishComponent	PublishComponent
csmsiRadioButton	RadioButton
csmsiRegistry	Registry
csmsiRegLocator	RegLocator
csmsiRemoveFile	RemoveFile
csmsiRemoveIniFile	RemoveIniFile
csmsiRemoveRegistry	RemoveRegistry
csmsiReserveCost	ReserveCost
csmsiSelfReg	SelfReg

Table 6-87 • MSI Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsiServiceControl	ServiceControl
csmsiServiceInstall	ServiceInstall
csmsiSFPCatalog	SFPCatalog
csmsiShortcut	Shortcut
csmsiSignature	Signature
csmsiTextStyle	TextStyle
csmsiTypeLib	TypeLib
csmsiUIText	UIText
csmsiUpgrade	Upgrade
csmsiVerb	Verb
csmsisys_Columns	_Columns
csmsisys_Tables	_Tables
csmsisys_Validation	_Validation

MSM Tables

An Application Catalog contains the following MSM tables:

Table 6-88 • MSM Tables

Table Name	Corresponding Windows Installer Table
csmsmActionText	ActionText
csmsmAdminExecuteSequence	AdminExecuteSequence
csmsmAdminUISequence	AdminUISequence
csmsmAdvtExecuteSequence	AdvtExecuteSequence
csmsmAdvtUISequence	AdvtUISequence
csmsmAppId	AppId

Table 6-88 • MSM Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsmAppSearch	AppSearch
csmsmBBControl	BBControl
csmsmBillboard	Billboard
csmsmBinary	Binary
csmsmBindImage	BindImage
csmsmCCPSearch	CCPSearch
csmsmCheckBox	CheckBox
csmsmClass	Class
csmsmComboBox	ComboBox
csmsmCompLocator	CompLocator
csmsmComplus	Complus
csmsmComponent	Component
csmsmCondition	Condition
csmsmControl	Control
csmsmControlCondition	ControlCondition
csmsmControlEvents	ControlEvents
csmsmCreateFolder	CreateFolder
csmsmCustomAction	CustomAction
csmsmDialog	Dialog
csmsmDirectory	Directory
csmsmDrLocator	DrLocator
csmsmDuplicateFile	DuplicateFile
csmsmEnvironment	Environment

Table 6-88 • MSM Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsmError	Error
csmsmEventMapping	EventMapping
csmsmExtension	Extension
csmsmFeature	Feature
csmsmFeatureComponents	FeatureComponents
csmsmFile	File
csmsmFileSFPCatalog	FileSFPCatalog
csmsmFont	Font
csmsmIcon	Icon
csmsmIniFile	IniFile
csmsmIniLocator	IniLocator
csmsmInstallExecuteSequence	InstallExecuteSequence
csmsmInstallUISequence	InstallUISequence
csmsmIsolatedComponent	IsolatedComponent
csmsmLaunchCondition	LaunchCondition
csmsmListBox	ListBox
csmsmListView	ListView
csmsmLockPermissions	LockPermissions
csmsmMedia	Media
csmsmMIME	MIME
csmsmModuleAdminExecuteSequence	ModuleAdminExecuteSequence
csmsmModuleAdminUISequence	ModuleAdminUISequence
csmsmModuleAdvtExecuteSequence	ModuleAdvtExecuteSequence

Table 6-88 • MSM Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsmModuleComponents	ModuleComponents
csmsmModuleConfiguration	ModuleConfiguration
csmsmModuleDependency	ModuleDependency
csmsmModuleExclusion	ModuleExclusion
csmsmModuleIgnoreTable	ModuleIgnoreTable
csmsmModuleInstallExecuteSequence	ModuleInstallExecuteSequence
csmsmModuleInstallUISequence	ModuleInstallUISequence
csmsmModuleSignature	ModuleSignature
csmsmModuleSubstitution	ModuleSubstitution
csmsmMoveFile	MoveFile
csmsmMsiAssembly	MsiAssembly
csmsmMsiAssemblyName	MsiAssemblyName
csmsmMsiDigitalCertificate	MsiDigitalCertificate
csmsmMsiDigitalSignature	MsiDigitalSignature
csmsmMsiFileHash	MsiFileHash
csmsmMsiPatchHeaders	MsiPatchHeaders
csmsmODBCAttribute	ODBCAttribute
csmsmODBCDataSource	ODBCDataSource
csmsmODBCDriver	ODBCDriver
csmsmODBCSourceAttribute	ODBCSourceAttribute
csmsmODBCTranslator	ODBCTranslator
csmsmPatch	Patch
csmsmPatchPackage	PatchPackage

Table 6-88 • MSM Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsmProgId	ProgId
csmsmProperty	Property
csmsmPublishComponent	PublishComponent
csmsmRadioButton	RadioButton
csmsmRegistry	Registry
csmsmRegLocator	RegLocator
csmsmRemoveFile	RemoveFile
csmsmRemoveIniFile	RemoveIniFile
csmsmRemoveRegistry	RemoveRegistry
csmsmReserveCost	ReserveCost
csmsmSelfReg	SelfReg
csmsmServiceControl	ServiceControl
csmsmServiceInstall	ServiceInstall
csmsmSFPCatalog	SFPCatalog
csmsmShortcut	Shortcut
csmsmSignature	Signature
csmsmTextStyle	TextStyle
csmsmTypeLib	TypeLib
csmsmUIText	UIText
csmsmUpgrade	Upgrade
csmsmVerb	Verb
csmsmsys_Columns	_Columns
csmsmsys_Tables	_Tables

Table 6-88 • MSM Tables (cont.)

Table Name	Corresponding Windows Installer Table
csmsmsys_Validation	_Validation

OS Snapshot Tables

An Application Catalog contains the following OS Snapshot tables:

Table 6-89 • OS Snapshot Tables

Table Name	Corresponding Windows Installer Table
csoscFile	N/A
csoscIniFile	N/A
csoscShortcut	N/A
csoscRegistry	N/A

Workflow Tables

An Application Catalog contains the following Workflow tables:

Table 6-90 • Workflow Tables

Table Name	Corresponding Windows Installer Table
wftblCommandLines	N/A
wftblTemplateTasks	N/A
wftblTemplateWorkflow	N/A
wftblTemplates	N/A
wftblTools	N/A
wftblUserWorkflows	N/A
wftblWorkflowTasks	N/A
wftblGroups	N/A
wftblGroupTools	N/A

Security Tables

An Application Catalog contains the following Security tables:

Table 6-91 • Security Tables

Table Name	Corresponding Windows Installer Table
assysRolePermissions	N/A
assysRoles	N/A
assysRoleTools	N/A
assysUserRoles	N/A

Workflow Manager Tables



Note • AdminStudio Workflow Manager is a Web-based application management system that has integrated functionality with AdminStudio.

An Application Catalog contains the following Workflow Manager tables:

Table 6-92 • AdminStudio Workflow Manager Tables

Table Name	Corresponding Windows Installer Table
asAMSAppRegistrations	N/A
AMS_AppAsgnRpt	N/A
AMS_AppAssignments	N/A
AMS_Application	N/A
AMS_ApplicationItem	N/A
AMS_AuditTrail	N/A
AMS_Company	N/A
AMS_CompanyBU	N/A
AMS_Contracts	N/A
AMS_DataMajorItemTpl	N/A

Table 6-92 • AdminStudio Workflow Manager Tables (cont.)

Table Name	Corresponding Windows Installer Table
AMS_DataMinorItemTextTpl	N/A
AMS_DataMinorItemTpl	N/A
AMS_Dictionary	N/A
AMS_EmailContent	N/A
AMS_Issues	N/A
AMS_ItemRole	N/A
AMS_KnowledgeCore	N/A
AMS_LanguageList	N/A
AMS_LanguageLongString	N/A
AMS_LanguageStrings	N/A
AMS_LanguageSymbols	N/A
AMS_LanguageUsages	N/A
AMS_Person	N/A
AMS_PersonRole	N/A
AMS_Policy	N/A
AMS_Preference	N/A
AMS_Queries	N/A
AMS_Roles	N/A
AMS_ServiceCompanyJoin	N/A
AMS_Templates	N/A
AMS_WFMajorItemTpl	N/A
AMS_WFMinorItemTpl	N/A
AMS_UploadedFiles	N/A

Application Catalog Replication Tables

When a Publication is published, not all of the tables from the original Application Catalog database are replicated on the Publisher Database Server. Click on one of the links below to see a list of included and excluded tables:

- [Tables Excluded From Replication](#)
- [Tables Included in Replication](#)

Tables Excluded From Replication

The following tables are excluded from Application Catalog Replication:

Table 6-93 • Tables Excluded from Replication

Description	Table Names
Pre-Existing Conflict Data	cstblConflictConfiguration cstblConflictACE cstblConflictPackages cstblConflictResults cstblConflictTransformList cstblConflictResultsDestPkgs
Database Security Tables	asssysRolePermissions asssysRoles asssysRoleTools asssysUserRoles
Custom Tables	All custom tables are excluded.
Miscellaneous	asAMSAppRegistrations asAMSWorkflowProjects cstblSubscribedDatabases cstblSubscribedData cstblPackageRelationships cstblIgnoreTableList cstblPackageHistoryLog cssysExtendedAttributeSchema cssysSchemaInfo cssysTables csmsisys_Validation csmsisys__Tables csmsisys__Columns tables wftblCommandLines wftblTemplateTasks wftblTemplateWorkflow wftblTemplates wftblTools wftblUserWorkflows wftblWorkflowTasks wftblGroups wftblGroupTools

Tables Included in Replication

One or both of the following sets of tables are included in Application Catalog Replication, depending upon the selections made on the **Publication Data Options** panel of the Publication Wizard: Groups and Package Data or Merge Module Data.

Table 6-94 • Tables Included in Replication

Tables	Description
General Package and Group Data	cstblPackage cstblGroups cstblGroupPackages cstblExtendedAttributeData cstblTransformFile cstblPatchFiles cstblSetupFiles All of the various csosc tables All of the various csmsi tables
Merge Module Tables	All of the various csmsm tables

cssys: AdminStudio System Tables

The following System tables are included in the AdminStudio Database Schema:

- [cssysExtendedAttributeSchema](#)
- [cssysSchemaInfo](#)
- [cssysTables](#)

cssysExtendedAttributeSchema

Extended attributes are optional attributes for packages, defined by an extended attribute description file (in XML format). Because you can manually create the description file, you have the flexibility to include information about each package that may be specific to your organization—such as the users or business groups that receive the package. The `cssysExtendedAttributeSchema` table contains data about the Database Schema's extended attributes. This table contains the following columns:

Table 6-95 • `cssysExtendedAttributeSchema` Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.

Table 6-95 • cssysExtendedAttributeSchema Columns

Column	Type	Key	Nullable	Description
Type	int		N	Indicates the data type of the extended attribute: 0 = Text, 1 = attached File, and 2 = value is specified by making a Selection from a list.
Name	text		N	The name of the extended attribute. The name can be up to 255 characters long.
DefaultValue	text		N	If the attribute type is Text, this column contains the default text value. If the attribute type is Selection, this column contains the list of possible selections. The value of this column can be up to 512 characters long.
FileFilter	text		N	This column contains the standard file filter used for file types. The value can be up to 255 characters long.
DefaultFileExtension	text		N	This column contains the standard file extension used to identify file types. The extension can be up to 255 characters long.
Caption	text		N	If the attribute type is File, the Caption is used for file types on the File Browse dialog box. The Caption can be up to 255 characters long.
HelpText	text		N	General help text displayed for Selection and Text types when changing the value of an extended attribute. Help text can be up to 512 characters long.

Remarks

This table is populated when the user changes the entries on the [Extended Attributes Tab](#) of the ConflictSolver and Application Manager [Options Dialog Box](#).

cssysSchemaInfo

The cssysSchemaInfo table contains data about the current Database Schema. This table contains the following columns:

Table 6-96 • cssysSchemaInfo Columns

Column	Type	Key	Nullable	Description
SchemaVersion	text		N	Readable text description of the schema version. This description can be up to 20 characters long.
UniqueID	int		N	Numeric identifier of the database version.
LastUser	text		Y	Not currently used.
LastUsed	datetime		Y	Not currently used.
ExtendedAttributesFile	text		Y	The file path to the Extended Attributes file that is selected on the Extended Attributes Tab of the ConflictSolver and Application Manager Options Dialog Box to 255 characters long.
ExtendedAttributesData	binary		Y	The actual Extended Attributes file.

cssysTables

The cssysTables table contains data about all ConflictSolver and Application Manager tables. This table contains the following columns:

Table 6-97 • cssysTables Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
TableName	text		N	The name of the table. This name can be up to 72 characters long.
DateCreated	datetime		N	Date the table was created.
TableType	int		N	Used internally.

Table 6-97 • cssysTables Columns

Column	Type	Key	Nullable	Description
MappedMsiTable	text		N	Name of the MSI table that is equivalent to this table. This table name can be up to 72 characters long.
Status	int		N	Not currently used.
BaseSchemaVersion	text		N	The version of the database schema that is associated with this table. The database schema version can be up to 20 characters long.
SchemaVersion	text		N	Not currently used.
DeleteMode	int		N	Used internally.

cstbl: AdminStudio Supporting Tables

The following supporting tables are included in the AdminStudio Database Schema:

Table 6-98 • cstbl: ConflictSolver Supporting Tables

Category	Table
Database Subscription Tables	<ul style="list-style-type: none"> • cstblSubscribedDatabases • cstblSubscribedData
Conflict Persistence Tables	<ul style="list-style-type: none"> • cstblConflictConfiguration • cstblConflictACE • cstblConflictPackages • cstblConflictResults • cstblConflictTransformList • cstblConflictResultsDestPkgs
Package and Transform File Tables	<ul style="list-style-type: none"> • cstblPackage • cstblTransformFile
Groups Table	<ul style="list-style-type: none"> • cstblGroups
Misc. Tables	<ul style="list-style-type: none"> • cstblPackageRelationships • cstblIgnoreTableList • cstblPackageHistoryLog
Extended Attributes Tables	<ul style="list-style-type: none"> • cstblExtendedAttributeData

Table 6-98 • cstbl: ConflictSolver Supporting Tables (cont.)

Category	Table
Group Packages	<ul style="list-style-type: none">cstblGroupPackages

cstblSubscribedDatabases

This table contains database connection information about currently subscribed databases. Each row of this table contains database connection information for one subscribed database. This table contains the following columns:

Table 6-99 • cstblSubscribedDatabases Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
ServerName	text		N	Name of the server where the subscribed database is located. This name can be up to 255 characters long.
DatabaseName	text		N	Name of the subscribed database. This name can be up to 255 characters long.
FriendlyName	text		N	A more readable form of the database connection information. It is displayed in the user interface. This name can be up to 255 characters long.
AuthMode	int		N	Used internally.
Flags	int		N	Used internally.
DatabaseType	int		N	Used internally.

cstblSubscribedData

This table contains the data that associates the packages in the subscribed and local databases. This table contains the following columns:

Table 6-100 • cstblSubscribedData Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
SubscribedDatabaseRowID_	int		N	Private Key identifier into a subscribed database entry.

Table 6-100 • cstblSubscribedData Columns

Column	Type	Key	Nullable	Description
Flags	int		N	Identifies the type of subscribed data.
LocalRowID_	int		N	Foreign Key identifier into the local database's cstblPackage table. This value identifies the imported package within the local database.
SubscribedRowID_	int		N	Foreign Key identifier into the subscribed database's cstblPackage table. This value identifies the Source package within the subscribed database.

cstblConflictConfiguration

This table contains principal information about a run of the [Conflict Wizard](#). This table contains the following columns:

Table 6-101 • cstblConflictConfiguration Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.
ProductName	text		N	Name of package that was tested. Name can be up to 255 characters long.
ProductVersion	text		N	Name of package version that was tested. Version can be up to 100 characters long.
PackagePath	text		Y	When performing an external package check, this column will contain the package path. Path can be up to 512 characters long.
ConflictRunDate	datetime		N	Date the run of the Conflict Wizard was performed.
NoOfErrors	int		N	Number of automatically-resolvable errors that were identified in this run of the Conflict Wizard.

Table 6-101 • cstblConflictConfiguration Columns

Column	Type	Key	Nullable	Description
NoOfWarnings	int		N	Number of warnings (i.e., not automatically resolvable) identified in this run of the Conflict Wizard.
NoOfPkgsChecked	int		N	Number of packages that were checked in this run of the Conflict Wizard.
NoResolved	int		N	Number of errors that were resolved since the last run of the Conflict Wizard over this package.
SaveToMST	int		N	Flag to indicate whether you want to save the conflict resolutions in a transform (.mst) file or in the original package.
MSTPath	text		Y	Path to the .mst file that will be used to save conflict resolutions.

Remarks

Each package has only the most recent set of conflict detection results persisted. In other words, each time the Conflict Wizard is run, the current conflict persistence data for this package is replaced by the new results.

cstblConflictACE

This table contains a list of ACEs that were used during the last run of the Conflict Wizard for this package. This table contains the following columns:

Table 6-102 • cstblConflictACE Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.
Name	text		N	Name of the ACE that was run. Name can be up to 72 characters long.
Category	text		Y	The category name of the ACE that was run. Category can be up to 72 characters long.

Table 6-102 • cstblConflictACE Columns

Column	Type	Key	Nullable	Description
Description	text		Y	Description of the ACE that was run. Description can be up to 512 characters long.

Remarks

This information is duplicated from the ACE file to use for display purposes in ConflictSolver and for the **Run Again** function in the Conflict View.

cstblConflictPackages

This table contains a list of packages that were used in the last run of the Conflict Wizard for this package. This table contains the following columns:

Table 6-103 • cstblConflictPackages Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.
LocalPkgRowID_	int		N	Foreign Key identifier into the Target package that was tested in the last run of the Conflict Wizard.
ProductName	text		N	Name of the package that was tested in the last run of the Conflict Wizard. Name can be up to 255 characters long.
Comments	text		Y	Comments describing the package that was tested in the last run of the Conflict Wizard. Comments can be up to 4000 characters long.
Flags	int		N	Used internally.

Remarks

This information is duplicated from the ACE file to use for display purposes in ConflictSolver and for the **Run Again** function in the Conflict View.

cstblConflictResults

For each identified conflict, an entry is made in this table. This table contains the following columns:

Table 6-104 • cstblConflictResults Columns

Column	Type	Key	Nullabl e	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.
SourceRowID_	int		N	Foreign Key identifier into the Source record that was identified as causing the conflict.
UpdateValue	text		N	For conflicts that have an automatic resolution, this column will contain data specific for that resolution. For non-resolvable ACEs, this field is blank. Value can be up to 1024 characters long.
CARDQuery	text		N	For conflicts that have an automatic resolution, this column will contain data specific for that resolution. For non-resolvable ACEs, this field is blank. Query can be up to 1024 characters long.
ACERule	text		N	The name of the ACE Rule that was run to identify this conflict. Rule can be up to 72 characters long.
CARDRule	text		Y	The name of the CARD Rule, if any, that can fix this conflict. If there is no CARD associated with the ACE, this column is blank. Rule can be up to 72 characters long.
DisplayString	text		N	Text displayed in the Conflict View prior to ACE resolution. String can be up to 2048 characters long.
ResolveString	text		N	Text displayed in the Conflict View after ACE resolution. String can be up to 2048 characters long.
TableName	text		Y	Name of table associated with the ACE that contains the conflicting data. Name can be up to 255 characters long.
Category	int		N	Category of the ACE. It matches an entry in the ACE file.
State	int		N	Identifies the status of the ACE: Resolved or Unresolved.

Table 6-104 • cstblConflictResults Columns

Column	Type	Key	Nullable	Description
TargetProductName	text		N	Name of the Target Package that resulted in this conflict being identified in this run of the Conflict Wizard. Name can be up to 255 characters long.

cstblConflictTransformList

When performing an external package check, this table contains a list of the external package's associated transforms. This table contains the following columns:

Table 6-105 • cstblConflictTransformList Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.
TransformPath	text		N	Path of the transform file used in this external package check. Path can be up to 512 characters long.

Remarks

This table is relevant only for external package checks.

cstblConflictResultsDestPkgs

This table contains Target package conflict information. This table contains the following columns:

Table 6-106 • cstblConflictResultsDestPkgs Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table to which these conflict persistence results are associated. If the value is -1, it indicates that the data is associated with an external package.

Table 6-106 • cstblConflictResultsDestPkgs Columns

Column	Type	Key	Nullable	Description
ConflictResultsRowID —	int		N	Foreign Key to the Conflict Results table entry.
DestRowID_	int		N	Foreign Key identifier into the Target record that was identified as causing the conflict.

cstblPackage

The cstblPackage table is the core table that contains information about imported packages. This table contains a unique Package ID (the RowID column) that is referenced by all other data tables. This table contains the following columns:

Table 6-107 • cstblPackage Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PackageCode	text		N	The value of the equivalent MSI PackageCode. Code can be up to 100 characters long.
ProductCode	text		N	The value of the equivalent MSI ProductCode. Code can be up to 100 characters long.
ProductVersion	text		N	The value of the equivalent MSI ProductVersion. Version can be up to 100 characters long.
UpgradeCode	text		N	The value of the equivalent MSI UpgradeCode. Code can be up to 100 characters long.
ProductName	text		N	The value of the equivalent MSI ProductName. Name can be up to 255 characters long.
Manufacturer	text		N	The value of the equivalent MSI Manufacturer. Name can be up to 255 characters long.
ProductLanguage	int		N	The value of the equivalent MSI ProductLanguage.
FileName	text		N	UNC path to the MSI package. Name can be up to 255 characters long.

Table 6-107 • cstblPackage Columns

Column	Type	Key	Nullabl e	Description
Comments	text		N	User description of the package. Comments can be up to 4000 characters long.
ImportDate	datetim e		N	Date package was imported.
FileModifiedDate	datetim e		N	Derived from the file system, this is the date that the package (MSI) file was last modified.
Flags	int		N	Used internally.
DisplayedProductName	text		N	Name of package that is displayed in the Application Manager and ConflictSolver Product View. Name can be up to 255 characters long.

cstblTransformFile

Contains a list of transforms associated with a package. This table contains the following columns:

Table 6-108 • cstblTransformFile Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the package associated with this transform file.
TransformFile	text		N	Name of transform file. File can be up to 255 characters long.
ProductLanguage	text		N	Name of language of product in string form. Language can be up to 128 characters long.

Remarks

This table is used only if the imported package includes transforms.

cstblGroups

Application Manager allows categorizing related packages using Groups. The cstblGroups table is used to store the information about Application Manager groups. Application Manager ships with one row in this table that defines the root node in the Groups tree. A group may contain imported packages or other sub-groups; these parent-child relationship details are stored in the [cstblGroupPackages](#) table. This table contains the following columns:

Table 6-109 • cstblGroups Columns

Column	Type	Key	Nullabl e	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
GroupName	text		N	Name of group. Name can be up to 72 characters long.
Description	text		N	Description of group. Description can be up to 128 characters long.
Comments	text		N	Comments associated with group. Comments can be up to 255 characters long.
Properties	int		N	Used internally.
CreatedDate	datetime		N	Date group was created.
ModifiedDate	datetime		N	Date group was last modified.

cstblPackageRelationships

Table that stores the relationship between packages which were considered duplicate packages at the time of importation. This table contains the following columns:

Table 6-110 • cstblPackageRelationships Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the first package associated with this duplicate relationship.
SecondPkgRowID_	int		N	Foreign Key identifier into the second package associated with this duplicate relationship.
Relationship	int		N	Used internally.

cstblIgnoreTableList

This table contains a list of tables to exclude during the import process. This table contains the following columns:

Table 6-111 • cstblIgnoreTableList Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
TableName	text		N	Name of ignored table. Name can be up to 255 characters long.
Comments	text		Y	Any user comments. Comments can be up to 4000 characters long.

cstblPackageHistoryLog

This table contains a log of changes made to a package. This table contains the following columns:

Table 6-112 • cstblPackageHistoryLog Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the package associated with this history log.
LogText	text		Y	Information specific for this type of log entry.
ShortDesc	text		Y	Short description of this type of log entry. Description can be up to 255 characters long.
LogData	image		Y	Not currently used.
Type	int		N	Indicates what type of log entry this represents.
CreateDate	datetime		N	Date the history log was created.
UserID	text		Y	Not used.

Remarks

Currently only used to store information about the import configuration options.

cstblExtendedAttributeData

This table contains the actual Extended Attribute data that is associated with a package. This table contains the following columns:

Table 6-113 • cstblExtendedAttributeData Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PkgRowID_	int		N	Foreign Key identifier into the package associated with this extended attribute data.
Name_	text		N	Private Key into the extended attribute schema table. Name can be up to 255 characters long.
DataField	text		Y	If extended attributes are of File type, this column contains the path to the file. If extended attributes are in Text or Selection type, this column contains the actual data value of the extended attributes. Data can be up to 255 characters long.
DataBLOBField	binary		Y	If the extended attribute is of File type, this column contains the contents of that file.

Remarks

Only if the user has entered extended attribute data for a package (an extended attribute element), would there be an entry in this table.

cstblGroupPackages

This table defines the relationship between packages and groups. This table contains the following columns:

Table 6-114 • cstblGroupPackages Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
ParentGroupCode	int		N	Foreign Key identifier into the cstblGroup table. Parent group containing either the package or the group entry (Group Row ID).

Table 6-114 • cstblGroupPackages Columns

Column	Type	Key	Nullable	Description
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table. If this is a group, the value in this column is 0. Otherwise, it contains the PackageRowID.
MemberGroupCode	int		N	Foreign Key identifier into the cstblGroup table. If this is a package, the value in this column is -1. Otherwise, it contains the GroupRowID.

as: Job Manager Tables

The following Job Manager tables are included in the AdminStudio Database Schema:

- [AsJob](#)
- [AsJobQueue](#)
- [AsJobStep](#)
- [AsJobStepMaster](#)
- [AsJobQueueStepDetail](#)
- [AsNotification](#)
- [AsQueueStatus](#)
- [AsSchedule](#)

AsJob

This table identifies all jobs within the Enterprise Server.

Table 6-115 • AsJob Columns

Column	Type	Key	Nullable	Description
JobId	Int	Y	N	The primary key to uniquely identify a job.
JobName	Varchar(32)	N	N	The user provided name for the job.
UserName	Varchar(255))	N	N	The user name that created this job and derived as part of the Enterprise Server login process.
JobType	Varchar(5)	N	N	A string identifying the two job types of Custom (cust) and Template (temp) jobs.

Table 6-115 • AsJob Columns (cont.)

Column	Type	Key	Nullable	Description
Created	Datetime	N	N	The time when the job was created. Generated automatically.
Deleted_Flag	Char(1)	N	N	Not currently used.
Deleted	Datetime	N	Y	Not currently used.
Description	Varchar(400)	N	Y	An extended description used to elaborate the purpose of a (template) job.

AsJobQueue

This table is the active queue for all job instances which are available to run, or which have run.

Table 6-116 • AsJobQueue Columns

Column	Type	Key	Nullable	Description
JobQueueId	Int	Y	N	The primary key to uniquely identify a job queue entry.
QueueName	Varchar(32)	N	Y	The name of the job queue entry.
LastAction	Varchar(200)	N	Y	The last status message generated by the job execution. Initially empty until the job actually runs.
Priority	Int	N	Y	Relative priority value giving preference to jobs scheduled for immediate execution
CompletionTime	DateTime	N	Y	The time that the last run of the job completed.
RequestTime	DateTime	N	N	The time after which the job will be set as available to run.
JobId_	Int	N	N	Foreign key into the AsJob table – JobId column. This value identifies which job is associated with this job queue entry.
UserName	Varchar(255)	N	N	The name of the user who added this job to the job queue and derived as part of the Enterprise Server login process.

Table 6-116 • AsJobQueue Columns (cont.)

Column	Type	Key	Nullable	Description
MetaData	Varchar(4096)	N	N	Job specific XML data used by template jobs.
StatusId_	Int	N	N	Foreign key into the AsQueueStatus table – StatusId column. This value identifies the status of this particular job queue entry.

AsJobStep

This table identifies the individual job steps for all jobs within the Enterprise Server.

Table 6-117 • AsJobStep Table Columns

Column	Type	Key	Nullable	Description
JobStepId	Int	Y	N	The primary key to uniquely identify a job.
JobStepName	Varchar(32)	N	N	The name of the job step.
UserName	Varchar(255)	N	N	The user name that created this job step and derived as part of the Enterprise Server login process.
MetaData	Varchar(4096)	N	N	Job step specific XML data
Created	Datetime	N	N	The time when the job step was created. Generated automatically.
Deleted_Flag	Char(1)	N	N	Not currently used.
Deleted	Datetime	N	Y	Not currently used.
Description	Varchar(400)	N	N	An extended description of the purpose of the job step.
JobId_	Int	N	N	Foreign key into the ASJob table indicating to which job this job step is associated.
JobOrder	Int	N	N	The ordinal number in which the job step is executed when the job itself is executed.
StepId_	Int	N	N	Foreign key into the AsJobStepMaster table which identifies the type of job.

AsJobStepMaster

This table contains the job step type identification information.

Table 6-118 • AsJobStepMaster Table Columns

Column	Type	Key	Nullable	Description
StepId_	Int	Y	N	The primary key identifier of particular job step type.
JobName	Varchar(30)	N	N	A specific name for a given job step type.
JobStepDescription	Varchar(100)	N	N	A descriptive name for the job step type.

AsJobQueueStepDetail

This table contains the individual job step statuses for the last run of any given job.

Table 6-119 • AsJobQueueStepDetail Table

Column	Type	Key	Nullable	Description
JobQueueId_	Int	Y	N	Primary key into the AsJobQueue table – JobQueueId column. Identifies the job queue entry which was executed to generate this job step status.
JobStepId_	Int	Y	N	Primary key into the AsJobStep table – JobStepId column. Identifies the particular job step.
RunOn	Datetime	N	Y	The time when this job step completed its execution.
LastOperation	Varchar(200)	N	N	The last message generated by this particular job step.
StatusId_	Int	N	N	Foreign key into the AsQueueStatus table – StatusId column. Describes the specific status of this particular job step.
ReportData	Varchar(4000)	N	N	Extended job step specific data which describes how the job was processed.

AsNotification

This table defines any message alerts associated with a job and sent via e-mail if the job is set to a particular status.

Table 6-120 • AsNotification Table Columns

Column	Type	Key	Nullabl e	Description
NotificationId_	Int	Y	N	The primary key identifier of particular job notification record.
SentTo	Varchar(400)	N	N	Self-explanatory
Subject	Varchar(100)	N	Y	Self-explanatory
Message	Varchar(800)	N	Y	Self-explanatory
StatusId_	Int	N	N	Foreign key into the AsQueueStatus table – StatusId column. Identifies the job status which is associated with this notification record.
JobQueueId –	Int	N	N	Foreign key into the AsJobQueue table – JobQueueId column. Identifies the job queue entry to which this notification is associated.

AsQueueStatus

This table contains the status ids for each state in which a job can be set to. Do not change this table.

Table 6-121 • AsQueueStatus Table Columns

Column	Type	Key	Nullabl e	Description
StatusId	Int	Y	N	The primary key identifier of particular job status value.
StatusName	Varchar(32)	N	N	A specific name for a given job status.
Description	Varchar(250)	N	N	A descriptive name for the job status

AsSchedule

This table contains scheduling information for any particular job. Optional. Only jobs with schedule information will have this table entry.

Table 6-122 • AsSchedule Table Columns

Column	Type	Key	Nullabl e	Description
JobQueueId_	Int	Y	N	Foreign key in the AsJobQueue table – JobQueueId column. The job for which this schedule information is associated.
ScheduleDate	Datetime	N	Y	Effectively, the time for which the job is currently scheduled to run.
IsRepeatable	Int	N	Y	A flag indicating if the job is reschedulable. 1 – yes
RecurrenceInterval	Int	N	Y	Identifies the daily, weekly, or monthly reoccurrence interval.
RepeatCount	Int	N	Y	Identifies the number of times to reschedule the job (Mutually exclusive with EndDate .)
EndDate	Datetime	N	Y	Identifies an absolute end time beyond which no further rescheduling will occur. (Mutually exclusive with RepeatCount .)

cspch: OS Security Patch Wizard Tables

The following OS Security Patch Wizard tables are included in the AdminStudio Database Schema:

- cspchBulletin
- cspchPatchFileData
- cspchPatchOperation
- cspchPatchPackage
- cspchPatchProducts
- cspchPatchRegistryData
- cspchProducts
- cspchProductServicePacks
- cspchServicePacks

pdt: Predeployment Test Tables

The following Predeployment Test tables are included in the AdminStudio Database Schema:

- pdtConflictMessages
- pdtConflictPackages
- pdtConflictTest
- pdtDeployTest
- pdtDeployTestMessages
- pdtJobInfo
- pdtJobMachineResult
- pdtMachineRoot
- pdtOverrideResults

Miscellaneous Tables

The following miscellaneous tables are included in the AdminStudio Database Schema:

- cstblPackageProperties
- cstblPackageVersionHistory
- cstblPatchProperties
- cstblTransformProperties
- sysManagedCatalog
- cssysSoftRepAttributes

Application Manager Command-Line Functionality



Note • If you invoke Application Manager using the command line `ismide.exe` without passing any parameters, Application Manager will launch in UI mode.



Tip • By default, all packages are imported into the root group. If you want to import packages to specific groups, you must use a configuration file (`-C` parameter), and specify a group for each package.

ConflictSolver and Application Manager support the following command-line parameters. These parameters are case-insensitive, and must be preceded with either a dash symbol (-) or a forward slash (/).

Table 6-123 • Command Line Parameters

Command Line Parameter & Examples	Description
-? iscmide.exe -?	Displays version information and help text for command-line parameters.
-C"configuration_file_name" iscmide.exe -I -C"C:\MyConfigs\myconfig.ini"	The name and location of a configuration file (.ini) containing any required parameters for import. If you are using a configuration file, the only necessary parameters to pass at the command line are -I and -C"configuration_file_name". You can include all other parameters inside the INI file.
-D"application_catalog_name" iscmide.exe -S"mysql\sql1" -U"admin" -P"admin" -D"mycatalog" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mytrans.mst;"	The name of the Application Catalog. This parameter is only used for SQL Server Application Catalogs, and is a required parameter when using a SQL Server-based Application Catalog.
-I iscmide.exe -I -C"c:\mypackages\myconfig.ini"	This option indicates that a bulk import operation is to be performed. If you are using -IF or -IMM, you do not need to specify this, as these parameters inform Application Manager to perform an import operation.
-IF"msi_file_name[; mst1; mst2...]" -IF"msi_file_name[; msp1; msp2...]" -IF"msi_file_name[; mst1; msp1; msp2...]" iscmide.exe -F"c:\mycatalogs\conflict.mdb" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mytrans.mst;" iscmide.exe -F"c:\mycatalogs\conflict.mdb" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mypatch.msp; c:\mypackages\mytrans.mst;" iscmide.exe -F"c:\mycatalogs\conflict.mdb" -IF"c:\mypackages\mymsi.msi	The name and full path of the MSI file to be imported, and optionally a semicolon-delimited list of transforms or patches to be applied before importing the package.

Table 6-123 • Command Line Parameters

Command Line Parameter & Examples	Description
-IMM"merge_module_file_name" iscmide.exe -F"c:\mycatalogs\conflict.mdb" -IMM"c:\mymodules\crystal.msm"	The name and full path of the merge module to be imported.
-INCP iscmide.exe -INCP"C:\Acrobat.ncp"	Use to specify the Native Channel Packager (NCP) file that is to be imported.
-L"logfile_name" iscmide.exe -I -C"c:\mypackages\myconfig.ini" -L"c:\mypackages\mylog.txt"	The output log file name.
-P"password" iscmide.exe -S"mysql\sql1" -U"admin" -P"admin" -D"mycatalog" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mytrans.mst;"	<p>The password for the SQL Server Application Catalog.</p> <p>This parameter is only used for SQL Server Application Catalogs, and is a required parameter when using a SQL Server-based Application Catalog. The only exception to this requirement is if you want to take advantage of AdminStudio's integrated security. In this case, do not use the -P parameter.</p>
-Q iscmide.exe -Q	Starts Application Manager as a system tray icon application without showing the full Application Manager Interface. When this option is specified with bulk import options, Application Manager exits once import is complete.
-S"server_name[instance_name]" iscmide.exe -S"mysql\sql1" -U"admin" -P"admin" -D"mycatalog" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mytrans.mst;"	<p>The SQL Server where the Application Catalog resides. If this parameter is passed, the -F parameter is ignored.</p> <p>This parameter is only used for SQL Server Application Catalogs, and is a required parameter when using a SQL Server-based Application Catalog.</p>

Table 6-123 • Command Line Parameters

Command Line Parameter & Examples	Description
-U"login_id" iscmide.exe -S"mysql\sql1" -U"admin" -P"admin" -D"mycatalog" -IF"c:\mypackages\mymsi.msi; c:\mypackages\mytrans.mst;"	The user name for the SQL Server Application Catalog This parameter is only used for SQL Server Application Catalogs, and is a required parameter when using a SQL Server-based Application Catalog. The only exception to this requirement is if you want to take advantage of AdminStudio's integrated security. In this case, do not use the -U parameter.

Specifying User Interface Mode via Command Line

There are three modes of operation for ConflictSolver: Application Manager, ConflictSolver, and ConflictSolver Process Assistant. The following Command Line options launch ConflictSolver in these modes of operation.

Table 6-124 • User Interface Mode Command Line Options

Option	Description
-A	Process Assistant Mode
-CS	ConflictSolver Mode
-APP	Application Manager Mode

Using a Configuration File

This section explains how to define a configuration file to use during the import process and how to use a configuration file with command-line options. The following topics are included:

- [Application Manager Configuration File](#)
- [Using a Configuration File with Command-Line Options](#)

Application Manager Configuration File

In addition to supporting individual command-line parameters, Application Manager can also use a configuration file (when specified using the -C"configuration_file_name" parameter). This INI file can contain the values for all required parameters during the import process.

Configuration File Examples

The following examples are provided:

- [Configuration File for SQL Server Application Catalog for a Named User](#)
- [Configuration File for SQL Server Application Catalog for a Trusted User](#)
- [Configuration File for Applying Transforms and/or Patches During Command-Line Import](#)

Configuration File for SQL Server Application Catalog for a Named User

```
[General]
DatabaseType=SQL
LogFile=c:\temp\importlog.txt
MSIFile=3
MSMFile=1

[SQL]
Server=ConflictSolverSQL2K
UserID=Admin
Password=mypassword
Database=AdminStudio70

[MSIFile-1]
File=\\server\Data1.msi
Transform1=\\server\Data1a.mst
Transform2=\\server\Data1b.mst
Group=OfficeApps

[MSIFile-2]
File=\\server\Data2.msi
Group=OfficeApps\Secondary

[MSIFile-3]
File=\\server\Data3.msi

[MSMFile-1]
File=\\server\CrystalReports.msm
```

Configuration File for SQL Server Application Catalog for a Trusted User

```
[General]
DatabaseType=SQL
LogFile=c:\temp\importlog.txt
MSIFile=3
MSMFile=1

[SQL]
Server=ConflictSolverSQL2K
Database=AdminStudio70

[MSIFile-1]
File=\\server\Data1.msi
Transform1=\\server\Data1a.mst
Transform2=\\server\Data1b.mst
Group=OfficeApps
```

```
[MSIFile-2]
File=\\server\Data2.msi
Group=OfficeApps\Secondary[MSIFile-3]
File=\\server\Data3.msi[MSMFile-1]
File=\\server\CrystalReports.msm
```

Configuration File for Applying Transforms and/or Patches During Command-Line Import

```
[General]
DatabaseType=Access
LogFile=c:\temp\importlog.txt
MSIFile=3
MSMFile=1
OtherSetupFile=1

[MSIFile-1]
File=\\server\Data1.MSI
Transform1=\\server\Data1a.MST
Transform2=\\server\Data1b.MST
Patch1=\\server\Data1p.MSP
Patch2=\\server\Data2p.MSP
AdminInstallLocation=\\Server\Shared\Data1

[MSIFile-2]
File=\\server\Data2.MSI
Patch1=\\server\Data1p.MSP
AdminInstallLocation=\\Server\Shared\Data2


[MSIFile-3]
File=\\server\Data3.MSI
Transform1=\\server\Data3a.MST
Patch1=\\server\Data1p.MSP
AdminInstallLocation=\\Server\Shared\Data3

[OtherSetupFile-1]
SetupName=AdminStudio for Macintosh
SetupDirectory=C:\AdminStudio\MacFiles
FullDirectory=1
```

Parameter Explanation

[General] Section

Table 6-125 • Parameter Explanation: [General] Section

Parameter	Description
LogFile	Using this parameter, you can specify the name and location of the output log file. The LogFile parameter corresponds to the -L command-line parameter.
MSIFile	Use this parameter to indicate the number of MSI files to be imported. Each MSI file is denoted in subsequent INI file sections. If you are only importing merge modules, you can omit this parameter from your configuration file.
MSMFile	Use this optional parameter to indicate the number of merge modules to be imported. Each merge module file is denoted in subsequent INI file sections.  Note • If you are only importing Windows Installer packages, you can omit this parameter from your configuration file.

[SQL] Section

Table 6-126 • Parameter Explanation: [SQL] Section

Parameter	Description
Server	Using this required parameter, you can provide the name of the SQL Server. It corresponds to the -S command-line parameter.
UserID	Using this parameter, you can provide the login name for the SQL Server. It corresponds to the -U command-line parameter. This is required for non-trusted logins.
Password	Using this parameter, you can provide the password for the SQL Server. It corresponds to the -P command-line parameter. This is required for non-trusted logins.
Database	Use the Database parameter to provide the catalog name for the SQL Server. It corresponds to the -D command-line parameter. If a value is not specified, the default SQL Server Application Catalog for the specified login will be used.

[MSIFile-n] Section

Each Windows Installer package to be imported into the Application Catalog must be described in its own section, numbered sequentially ([MSIFile-1], [MSIFile-2], etc.). Each section must contain the name and location of the file, and any transforms or patches to apply to the file prior to import.

Table 6-127 • Parameter Explanation: [MSIFile-n] Section

Parameter	Description
File	Using the File parameter, you can specify name and location of the Windows Installer package (.msi) you are importing. This parameter is required.
Transformn	Use this parameter to specify the name and location of a transform to apply to the Windows Installer package prior to import. Each subsequent transform increases the value of <i>n</i> (Transform1, Transform2, Transform3, etc.).
Patchn	Use this parameter to specify the name and location of a patch to apply to the Windows Installer package prior to import. Each subsequent patch increases the value of <i>n</i> (Patch1, Patch2, Patch3, etc.).
AdminInstallLocation	When applying a patch to an MSI package, it is necessary to perform an Administrative install of the MSI package and then perform an Administrative install of each patch package one by one. Use this parameter to specify the location where the Administrative install will be performed.
Group	You can use the Group parameter to specify the group into which the package should be imported. Use a “\” to indicate a group hierarchy. If no group is specified, the package is imported into the root group.
SetupName	Identifies the name of the imported setup package (an Other Setup Type).
SetupDirectory	Identifies the location of the Other Setup Type setup files.
FullDirectory	Specifies whether to import files in the selected directory and all subdirectories (1) or just the files in the selected directory (0).

[MSMFile-n] Section

Each merge module to be imported into the Application Catalog must be described in its own section, numbered sequentially ([MSMFile-1], [MSMFile-2], etc.). Each section must contain the name and location of the file.

Table 6-128 • Parameter Explanation: [MSMFile-n] Section

Parameter	Description
File	Use this parameter to specify the name and location of the merge module (.msm) you are importing.

[NCPFile-n] Section

Each Native Channel Packager (NCP) file to be imported into the Application Catalog must be described in its own section, numbered sequentially ([NCPFile-1], [NCPFile-2], etc.). Each section must contain the name and location of the file.

Table 6-129 • Parameter Explanation: [NCPFile-n] Section

Parameter	Description
File	Use this parameter to specify the name and location of the NCP file (.ncp) you are importing.

Using a Configuration File with Command-Line Options

Application Manager supports use of a configuration file to pass parameters during command-line import. This is extremely useful if you are importing multiple packages and/or merge modules simultaneously. Use the following command line to use a configuration file:

```
ismide.exe -C"configuration_file_name"
```

Replace **configuration_file_name** with the name and location of the configuration file to use.

Importing

This section includes topics on how to perform import operations from the command line. The following topics are included:

- [Applying Transforms and Patches During Command-Line Import](#)
- [Importing Multiple Windows Installer Packages Simultaneously](#)
- [Importing Multiple Merge Modules Simultaneously](#)
- [Simultaneously Importing Windows Installer Packages and Merge Modules](#)
- [Using the Command Line to Import All Packages in a Directory](#)
- [Running Import Silently](#)
- [Creating a Log File During Command-Line Import](#)

Applying Transforms and Patches During Command-Line Import

Application Manager provides two mechanisms for applying transforms or patches to a Windows Installer package during import. First, if you are only importing a single package into Application Manager from the command line, you can pass transforms or patches after the file name, as in the following command lines:

Table 6-130 • Applying Transforms and Patches During Import

Function	Command
TRANSFORM:	<code>iscmide.exe -IF"msi_file;[mst1;mst2;]"</code>
PATCH:	<code>iscmide.exe -IF"msi_file;[msp1;msp2;]"</code>
BOTH:	<code>iscmide.exe -IF"msi_file;[mst1;msp1;]"</code>

Replace `msi_file` with the name and location of the Windows Installer package, and `mst1` and `mst2` or `msp1` and `msp2` with the names and locations of the transforms or patches to apply. For example, if your MSI file was named `data1.msi` and your two MST files were named `alpha.mst` and `gamma.mst`, your command line would look like the following:

```
iscmide.exe -IF"data1.msi;alpha.mst;gamma.mst;
```



Note • Depending on the situation, it may be necessary to pass additional command-line parameters.

If you are importing multiple Windows Installer packages, and applying transforms or patches to them, use a configuration file in which you can specify the names and locations of the packages and associated transforms or patches. Use the following command line to use a configuration file:

```
iscmide.exe -C"configuration_file_name"
```

Replace `configuration_file_name` with the name and location of the configuration file to use.

Importing Multiple Windows Installer Packages Simultaneously

The best way of importing multiple Windows Installer packages simultaneously into an Application Catalog is via a configuration file. Use the following command line to call your configuration file, in which you can specify multiple Windows Installer packages to import.

```
iscmide.exe -C"configuration_file"
```

Replace `configuration_file` with the name and location of the configuration file containing the names and locations of the packages you want to import.

Importing Multiple Merge Modules Simultaneously

The best way of importing multiple merge modules simultaneously into an Application Catalog is via a configuration file. Use the following command line to call your configuration file, in which you can specify multiple merge modules to import.

```
ismide.exe -C"configuration_file"
```

Replace `configuration_file` with the name and location of the configuration file containing the names and locations of the merge modules you want to import.

Simultaneously Importing Windows Installer Packages and Merge Modules

To simultaneously import both Windows Installer packages and merge modules into an Application Catalog, use a configuration file from the command line. In this configuration file, you can specify the names and locations of both merge modules and Windows Installer packages you want to import.

Use the following command-line to specify your configuration file:

```
ismide.exe -C"configuration_file_name"
```

Replace `configuration_file_name` with the name and location of the configuration file to use.

Using the Command Line to Import All Packages in a Directory

It is possible from the command line to import all packages in a directory. This can be done by placing the following line in a batch file (modifying it as necessary for your specific environment):

```
for %%a in (*.msi) do ISCMIDE  
-S"server_name"  
-U"userid"  
-P"password"  
-D"Application_Catalog_Name"  
-IF"%%a"  
-Q  
-L"%%aLog.txt"
```

The above statement runs Application Manager, importing each MSI package in the Application Catalog, and creates a log file with the name of the .msi file prepended to the log file name.

However, the above command makes the following assumptions:

- ISCMIDE.exe is in the path; otherwise, include the full path to ISCMIDE.exe.
- All packages are being imported into the specified SQL Server Application Catalog (provide the server name, userid, password, and Application Catalog name with the appropriate parameters -S, -U, -P, and -D, respectively).
- The above command will start Application Manager, import a file, exit Application Manager, and then restart Application Manager for each MSI file. If this is not acceptable, use a configuration file instead.
- No transforms are applied to the imported packages. If you need to apply transforms, use a configuration file instead.

- Application Manager starts in quiet mode. If it is switched to full mode, it is your responsibility to end the Application Manager process so the next file can be imported.

Running Import Silently

Application Manager can be run silently in system tray mode by using the following command line:

```
ismide.exe -Q
```

An icon for Application Manager, with a corresponding context menu available by right-clicking on the icon, appears in the system tray. If you pass the -Q parameter in addition to other parameters, progress is displayed in the tool tip available when you mouse over the icon. This is beneficial if you are importing a sizeable amount of packages, yet want to monitor the progress periodically.

In quiet mode, when import from the command line ceases, Application Manager automatically exits.

Creating a Log File During Command-Line Import

It may be necessary or beneficial to create a log file during package/merge module import from the command line. You may want to see the results of the import, or determine why certain packages were not processed as expected. To create a log file, use the following command-line:

```
ismide.exe -L"log_file_name"
```

Replace `log_file_name` with the name and location of the log file you want to create.



Note • It may be necessary to pass additional command-line parameters, depending on the task you are performing from the command line.

Connecting to Standalone Application Catalogs

This section includes the following topics:

- [Connecting to a Specific Standalone Application Catalog Using Command-Line Options](#)
- [Creating Shortcuts to Specific Standalone Application Catalogs](#)

Connecting to a Specific Standalone Application Catalog Using Command-Line Options

Using the following command line, you can connect to a specific SQL Server Application Catalog:

```
ismide.exe -S"sql_server_name" -U"user_id" -P"password"
```

Replace `sql_server_name` with the specific SQL Server name (including path information). Replace `user_id` and `password` with the UserID and password for the server.



Note • Depending on the Application Catalog configuration, it may be necessary to pass additional command-line parameters, such as the Application Catalog name.

Creating Shortcuts to Specific Standalone Application Catalogs

For convenience, you may want to create a Windows shortcut to launch ConflictSolver and automatically connect to a specific Application Catalog.



Task: To create the shortcut:

1. On the Windows Desktop, right-click, point to New, and select Shortcut. The Create Shortcut Wizard appears.
2. Click the Browse button and navigate to the ConflictSolver executable (ISCMIDE.exe). Typically, it is in the *AdminStudio Installation Directory\ConflictSolver* directory. When you have located it, click OK and then click Next.
3. Type a name for the shortcut and click Finish. The new shortcut appears on the Desktop.
4. Right-click on the shortcut and select Properties.
5. In the **Properties** dialog box, append the necessary command-line parameters to the end of the target, such as:

`iscmide.exe -S"sql_server_name" -U"user_id" -P"password"`
6. Click OK to dismiss the dialog box.



Note • If you are creating a shortcut to a Microsoft SQL Server Application Catalog, you need to provide values for the SQL Server name, UserID, password, and Application Catalog name using the -S or -O, -U, -P, and -D parameters. See [Application Manager Command-Line Functionality](#) or [Connecting to a Specific Standalone Application Catalog Using Command-Line Options](#).

Application Catalog Replication Command Line Functionality



Note • These options are provided in case you would like to run Application Catalog Replication using the command line. However, this functionality can be invoked from the main AdminStudio interface.

The following Application Catalog Replication command-line options are supported by AdminStudio:

Table 6-131 • Replication Command Line Options

Option	Description
-c"Connection String"	Optional, if not present we attempt to connect to the default database. Otherwise, we will prompt the user for a connection of an existing database.
-q	Silent launch without splash dialog box.
-h[number]	Parent window handle. (Optional)

Part 3

Repackaging and Customizing Installations

This part of the AdminStudio 9.5 User Guide includes the following chapters:

- [Repackaging Legacy Installations Using the Repackaging Wizard](#)
- [Converting Legacy Installations Using the Repackager Interface](#)
- [Converting Windows Installer Packages to Virtual Applications](#)
- [Creating Customized Virtual Applications](#)
- [Customizing and Authoring Installations Using InstallShield](#)
- [Customizing Installations Using Tuner](#)

Repackaging Legacy Installations Using the Repackaging Wizard

Installations created for the Windows Installer service dramatically differ from traditional installations, making reusing legacy installations impossible without using a repackaging tool. You can use Repackager's Repackaging Wizard to capture the data placed on your system during installation and convert it into a Windows Installer (.msi) package, which you can then customize and distribute according to your organization's needs.

Documentation regarding using the Repackaging Wizard is presented in the following sections:

Table 7-1 • Using the Repackaging Wizard

Section	Description
About Repackaging	Introduces you to repackaging, explains various repackaging methods, lists Repackaging Best Practices, explains how to include the InstallScript Engine with a Windows installer package, and reviews Repackager options.
Repackaging Methods	Describes the methods of repackaging that the Repackaging Wizard supports.
Configuring Repackager to Ensure Optimal Installation Capture	Describes how to configure Repackager in order to get optimal results when capturing an installation.
Repackaging Legacy Installations Using the Repackaging Wizard	Explains how to use the Repackaging Wizard to convert the following installations: <ul style="list-style-type: none">• InstallShield Professional 1.x to 5.1.x• InstallShield Professional 5.5 to 7.x• InstallShield InstallScript MSI• InstallShield DevStudio 9.x InstallScript• InstallShield Editor InstallScript

Table 7-1 • Using the Repackaging Wizard (cont.)

Section	Description
Repackaging Wizard Reference	Describes each of the dialog boxes and Wizard panels that you might encounter when using the Repackaging Wizard.



Note • For information on other Repackager features, see [Converting Legacy Installations Using the Repackager Interface](#) and [Converting Windows Installer Packages to Virtual Applications](#).

About Repackaging

This section introduces you to repackaging, lists Repackaging Best Practices, and explains how to set Repackager options.

- [Purpose of Repackaging Applications](#)
- [Supported Legacy Installation Types](#)
- [Repackaging 64-Bit Applications](#)
- [Repackaging Options Comparison](#)
- [Repackaging Wizard Best Practices](#)
- [About Repackaging on Clean Systems](#)
- [Including the InstallScript Engine With a Windows Installer Package](#)

Purpose of Repackaging Applications

Installations created for the Windows Installer service dramatically differ from traditional installations, making reusing legacy installations impossible without using a repackaging tool. Repackager assists you by capturing the data placed on your system during installation and converting it into a Windows Installer (.msi) package, which you can then customize and distribute according to your organization's needs.

Repackaging an installation into a Windows Installer package provides the following benefits:


- **Can customize it using InstallShield Editor or Tuner**—You can further configure or customize the Windows Installer package to meet your specific needs by editing the .msi file in InstallShield Editor or by creating transforms in InstallShield Editor or Tuner.
- **Can perform conflict analysis and resolution**—You can use ConflictSolver to check the Windows Installer package for conflicts that may exist between it and other known Windows Installer packages in an Application Catalog database, ensuring the proper installation and functioning of your installations.

- **Can implement application repair and feature advertising**—Finally, once converted to a Windows Installer package, the installation can take advantage of Windows Installer functionality such as application repair and feature advertisement.

Supported Legacy Installation Types

You can use both the Repackaging Wizard and the Repackager interface to create Repackager projects. The tool that you use depends upon the type of installation you are converting:



Table 7-2 • Methods of Creating Repackager Projects

Tool	Installation Source
Repackaging Wizard	<p>You can use the Repackaging Wizard to convert the following installations:</p> <ul style="list-style-type: none"> • InstallShield Professional 1.x to 5.1.x • InstallShield Professional 5.5 to 7.x • InstallShield InstallScript MSI • InstallShield DevStudio 9.x InstallScript • InstallShield Editor InstallScript <p>See Repackaging Legacy Installations Using the Repackaging Wizard.</p>
Repackager Interface	<p>You can use the Repackager interface to convert the following installations:</p> <ul style="list-style-type: none"> • Repackager 3.x output (.inc) • Microsoft SMS projects (.ipf) • Novell ZENworks 3.0, 3.1, or 4.0 projects (.axt/.aot) • WinINSTALL projects (.txt) (6.0, 6.5, 7.x) • Wise installation projects (.wse) • InstallShield Professional log files (.isl) <p>See Converting Legacy Installations Using the Repackager Interface.</p> <div>  </div> <hr/> <p><i>Edition •</i></p>

Repackaging 64-Bit Applications

Repackager has the capability to repackage both 32-bit and 64-bit applications, as well as hybrid applications (both 32-bit and 64-bit). The Repackaging Wizard remains a 32-bit application, but can be run on both 32-bit (x86) and 64-bit (x64) Windows operating systems. The following table lists the operating systems to use to repackage both 32-bit and 64-bit applications, and the operating systems those repackaged applications will run on.

Table 7-3 • Repackaging 32-Bit and 64-Bit Applications

Application Type	Repackage on ...	Will run on ...
64-bit application	Windows 64-bit OS  Note • When using the Repackaging Wizard to repackage a 64-bit application, you are required to use the Snapshot method; the Installation Monitoring method is disabled. For more information, see Snapshot Method .	Windows 64-bit OS
32-bit application	Windows 32-bit OS  Important • While it is possible to repackage a 32-bit application on a 64-bit OS, it is recommended that you use a 32-bit OS, to avoid inadvertently capturing any 64-bit data. If Repackager captures any 64-bit data, it will flag the package as a 64-bit application, meaning that it will only run on a 64-bit OS. See Excluding 64-Bit Data .	Windows 64-bit OS or Windows 32-bit OS



Excluding 64-Bit Data

It is strongly recommended that you repackage 32-bit applications on a 32-bit OS. However, if you choose to repackage a 32-bit application on a 64-bit OS, you need to make sure that you exclude any unnecessary 64-bit data, such as data from a 64-bit Windows Service that could be running or 64-bit files (stored in the System64Folder, ProgramFiles64Folder, or CommonFiles64Folder directories) or 64-bit registry entries (any entries stored in a node other than WOW6432Node).

Repackaging Options Comparison

The following table details the different options available to you when using Repackager, based upon source type, product and version:

Table 7-4 • Repackaging Options Comparison Chart

Source	Product / Version	Repackaging Method	Result
Media 	IS Professional 1.x to 5.1.x	Repackaging Wizard Installation Monitoring or Snapshot	Repackager project with no feature delineation
	IS Professional 5.5 to 7.x	Repackaging Wizard Installation Monitoring or Snapshot and SmartScan Wizard	Repackager project with feature delineation, including registry entries and shortcuts
	IS InstallScript MSI	Repackaging Wizard Installation Monitoring or Single Step Snapshot	Repackager project with feature delineation, including registry entries and shortcuts
	IS Editor InstallScript IS DevStudio 9.x InstallScript	Repackaging Wizard InstallShield Professional Logging Method	Repackager project with feature delineation, including registry entries and shortcuts. Also captures path variables and converts them into properties (using text substitution rather than hard-coded path names).
Project 	Repackager 3.x output (.inc) Microsoft SMS projects (.ipf) Novell ZENworks 3.0, 3.1, or 4.0 projects (.axt/.aot) WinINSTALL projects (.txt) (6.0, 6.5, 7.x) Wise installation projects (.wse)	Repackager Interface Select Open on the File menu to have Repackager automatically convert file to a Repackager project	Repackager project with no feature delineation
	InstallShield Editor Pro log files (.isl)	Repackager Interface Select Open on the File menu to have Repackager automatically convert file to a Repackager project	Repackager project with feature delineation

Once you have created a Repackager project, you can visually examine the files, .ini files, shortcuts, and registry data from the installation, and exclude any non-essential items. Then, you can build the Repackager project into an InstallShield Editor project (.ism) for further editing, or create a Windows Installer package (.msi).

Repackaging Wizard Best Practices

To ensure optimal performance of the Repackaging Wizard during repackaging and when working with Repackager projects, the following best practices are recommended:

- [Repackage on a Clean System](#)
- [Launch Repackager Remotely or Install Repackager on the Clean Machine](#)
- [Use the Repackager Interface to Exclude Unwanted Items](#)
- [Exit All Other Applications](#)
- [Only Repackage Non-Windows Installer Setups](#)

Repackage on a Clean System

It is essential that you repackage applications on a “clean” system to ensure you capture all changes made by the installation. A clean system typically consists of a computer with only the operating system and necessary service packs installed on it. Repackaging on a clean system provides the following benefits:

- **Prevents you from capturing Repackager files**—By repackaging on a clean system, you are ensuring that you do not inadvertently capture Repackager files during repackaging.
- **Ensures that you capture all of the necessary setup files**—If you do not repackage on a clean system, you may not capture all of the necessary files for the setup because the files may already be installed on the system.



Note • For more information, see [About Repackaging on Clean Systems](#).

Launch Repackager Remotely or Install Repackager on the Clean Machine

Because it is best to keep the number of packages installed on the clean machine to a minimum, you should launch Repackager remotely from the clean machine or install Repackager on the clean machine:

- **Launch Repackager Remotely**—You could install Repackager on a shared network drive and then launch Repackager remotely from the clean machine. See [Launching Repackager Remotely](#).
- **Install Repackager on clean machine**—You could install a copy of Repackager onto the clean machine. While it is preferable to launch Repackager remotely from the clean machine, if you do not have network access to an installation of the AdminStudio client tools, this is your best option. See [Installing Repackager on a Clean Machine](#).

Both of these options are explained in [Configuring Repackager to Ensure Optimal Installation Capture](#).

Use the Repackager Interface to Exclude Unwanted Items

You should repackage using the provided exclusions and then use the Repackager interface to visually remove unwanted items from the capture.

Because this occurs post-capture, you do not need to recapture the legacy setup if you inadvertently exclude items from the Windows Installer package you are building.



Note • Since Windows Installer does not support packaging device drivers, you would need to create Custom Actions to install device drivers. See [Using Custom Actions](#) in the Windows Installer help section for more information.

Exit All Other Applications

Other applications may lock files or directories, and may hinder the performance of the setup and repackaging. Therefore, exit all applications prior to repackaging.

Only Repackage Non-Windows Installer Setups

Windows Installer setups should not be repackaged. They should either be edited in InstallShield Editor, or, as Microsoft recommends, by creating a transform. This can be done using InstallShield Editor or Tuner.

You should not repackage Windows Installer (.msi) packages for the following reasons:

- Repackaging a Windows Installer package is against Microsoft Best Practices.
- If you make changes to a Windows Installer package, vendors will no longer provide support for that product.
- If you repackage a Windows Installer package, the component codes within the package are not retained and hence future patching or upgrades will not work.
- Traditionally, repackaging tools will ignore the Windows Installer-specific data in the Registry. This will result in an incomplete package.

Also, Repackager is not intended for repackaging operating system installations or service packs, or deeply integrated operating system components such as Internet Explorer. Moreover, components such as MDAC or DCOM should be included in the clean image, or installed by a setup using the vendor's redistributable.

About Repackaging on Clean Systems

For optimal results when using Repackager or OS Snapshot, you should perform these processes on a clean system. A clean system typically consists of a computer with only the operating system and necessary service packs installed on it. It is the baseline system that the computer requires to run.

Although it may be tempting to consider basic software, such as Microsoft Office, as part of the clean system, this can result in poor snapshots and repackaged setups. Each application you install on the baseline system adds to the DLLs, changes versions of files, makes new registry entries, etc. This may cause Repackager or the OS Snapshot Wizard to miss these during capture, which ultimately may lead to missing files or registry entries in repackaged setups, or unexpected conflicts between the operating system and Windows Installer packages.



Note • For more information on setting up a clean system to repackage on, see [Configuring Repackager to Ensure Optimal Installation Capture](#).

Alternate-Language Repackaging on Clean Machines

The standalone Repackager setup for clean machines does not install any language resources other than US English. Therefore, if you are Repackaging a setup on a clean system in a language other than in US English, you need to ensure you point to the correct template in the Repackaged Output View.

This can be on a mapped network drive, or you can copy the language-populated template (for example, ISProjBlankTpl.ism) to your clean system from the *AdminStudio Installation Directory\Editor\Support* directory.

Language-specific templates are available when you purchase InstallShield Editor Language Packs.

Including the InstallScript Engine With a Windows Installer Package


Should you need to include the InstallScript engine with your setup, all the major releases of the InstallScript engine are available in the *InstallScript_Engines* folder on the AdminStudio installation CD. For more information, see the *Update to the Latest InstallShield Installation Engines* Knowledge Base article at:

<http://consumer.installshield.com/kb.asp?id=Q108322>

Repackaging Methods

Repackager supports three methods of repackaging:

Table 7-5 • Repackaging Methods

Repackaging Method	Description
Installation Monitoring Method	<p>Repackager monitors system changes as an application is installed, and that data can be converted into a Windows Installer package. Installation Monitoring is the default method.</p>  <p><i>Edition •</i></p>
Snapshot Method	<p>Repackager compares a system snapshot before and after an installation, determines the changes that were made, and that data can be converted to a Windows Installer package.</p>
InstallShield Professional Logging Method	<p>Repackager reads logged output of InstallShield Editor and DevStudio 9.x InstallScript installations to obtain additional information that would not be captured by repackaging, such as path variables and a feature tree.</p>

Installation Monitoring Method

When using the **Installation Monitoring** method, the Repackaging Wizard monitors a system for any processes that are created during an installation. The Installation Monitoring method determines the dynamic interdependencies between files.

By monitoring these processes in the background, the Repackaging Wizard can identify files, shell links, and registry entries that are added, modified, or removed by the installation. The resulting files and Repackager output file can be converted into a Windows Installer package.

Installation Monitoring Method Considerations

Consider the following about Installation Monitoring when selecting a repackaging method:

- **Faster than Snapshot**—Installation Monitoring is significantly faster than the Snapshot repackaging method.
- **Clean system not required**—Although it is still a good practice to repackage on a clean system, it is not as important when using Installation Monitoring technology as it is when you use the Snapshot method.
- **Can exclude processes from the project**—When using the Installation Monitoring method, you can specify the processes that you want to exclude from the Installation Monitoring.

Snapshot Method

When using the **Snapshot** method, the Repackaging Wizard takes a reference snapshot of a system as a baseline configuration, performs the installation, and then takes a second snapshot.

The difference between the two snapshots is stored in a directory you specify, along with the Repackager output file (.inc). This file can then be converted into a Windows Installer package (.msi) using Repackager.

Snapshot Method Considerations

Consider the following about Snapshot technology when selecting a repackaging method:

- **Slower than Installation Monitoring**—The Snapshot method is significantly slower than the Installation Monitoring repackaging method.
- **Clean system is required**—When repackaging using Snapshot technology, you should use a clean system, with a baseline configuration for your target environment. If you do not repackage on a clean system, you may not capture all of the necessary files for the setup because the files may already be installed on the system.
- **Exclude anti-virus software directories**—Any machine that you use to repackage most likely has anti-virus software installed on it, even a “clean” machine. While you are repackaging an application, the real-time virus detection feature of anti-virus software could automatically update various cached files in its directories. In order to avoid repackaging errors when using the **Snapshot** method, you should exclude these directories. See [Excluding Directories and Subdirectories](#) for more information.



Note • Anti-virus software does not affect repackaging using the **Installation Monitoring** method.

InstallShield Professional Logging Method



Note • The InstallShield Professional Logging Method is supported for InstallShield Editor and DevStudio 9.x InstallScript installations.

Using the **InstallShield Professional Logging Method**, Repackager can read logged output of InstallShield Editor and DevStudio 9.x InstallScript installations. This enables Repackager to capture additional information from those installations that would not be captured by the **Installation Monitoring** or **Snapshot** methods: path variables and the feature tree.

Path Variables

When using the InstallShield Professional Logging Method, path variables are captured and converted into properties (using text substitution rather than hard-coded path names).

For example, if you were using a standard repackaging method, the path name of the application's executable file would be captured as follows:

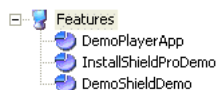
```
C:\Program Files\CompanyName\ApplicationName\ProgramName.exe
```

When using the InstallShield Professional Logging Method, the executable path name would be captured using a path variable instead of a hard-coded path name:

```
[INSTALLDIR]\ProgramName.exe
```

Feature Tree

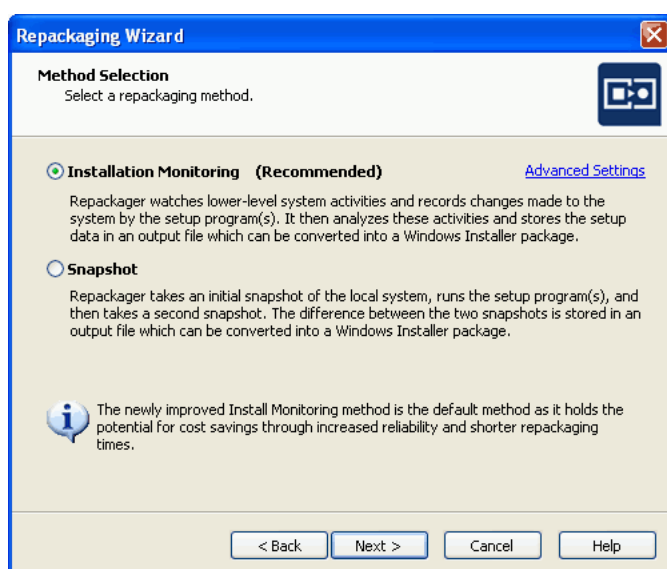
The InstallShield Professional Logging Method is able to group files, shortcuts and registry entries into features corresponding to InstallShield Professional components. Any items (files, folders, shortcuts, or registry entries) that are attached to a component directly or indirectly (through File Groups), are attached to corresponding features.



If you do not use the InstallShield Professional Logging Method to repackage an InstallShield Editor or DevStudio 9.x InstallScript installation, all files, shortcuts and registry entries would be installed together as one feature.

Selecting the InstallShield Professional Logging Method

You access the InstallShield Professional Logging Method through the Repackaging Wizard, but **InstallShield Professional Logging Method** is *not* offered as a choice on the **Method Selection Panel** of the Repackaging Wizard. Only **Installation Monitoring** and **Snapshot** are listed:



When you are repackaging an InstallShield Editor or DevStudio 9.x InstallScript installation and want to use the InstallShield Professional Logging Method, you can select either **Installation Monitoring** or **Snapshot** on the **Method Selection Panel**. However, if you select the **Snapshot** method, you must select **Single Step** on the **Snapshot Method Panel** (which would open next).



Caution • If you select the *Snapshot / Multiple Steps* repackaging method, Repackager will not recognize the setup as an InstallShield Editor or DevStudio 9.x InstallScript installation and the InstallShield Professional Setup Panel will not open.

After you select the **Installation Monitoring** or **Snapshot** method, and then specify an InstallShield Editor or DevStudio 9.x InstallScript installation on the Repackaging Wizard's Collect Product Information Panel, Repackager will automatically detect the InstallScript installation and will display the InstallShield Professional Setup Panel. On this panel, you can choose to use the **InstallShield Professional Logging Method** instead of the repackaging method you chose on the Method Selection Panel.

For instructions on how to use the InstallShield Professional Logging Method when repackaging InstallShield Editor or DevStudio 9.x InstallScript installations, see [Repackaging Using the InstallShield Professional Logging Method](#) and [InstallShield Professional Setup Panel](#).

Configuring Repackager to Ensure Optimal Installation Capture

Both repackaging methods, **Installation Monitoring** and **Snapshot**, involve installing an application and recording the system changes made by that installation. To ensure that you capture *all* changes made by the installation, you should, ideally, install the application onto a “clean machine” (a computer with only the operating system installed), as described in [About Repackaging on Clean Systems](#).

Depending upon your network connectivity, you should configure Repackager on a clean machine in one of the following ways:

Table 7-6 • Methods to Configure Repackager

Repackager Configuration	Description
Launching Repackager Remotely	If you have connectivity from a clean machine to a computer or network location that contains an installation of Repackager, you should launch Repackager remotely.
Installing Repackager on a Clean Machine	If you do not have any network connectivity on the clean machine, you should install Repackager on the clean machine.



Note • You cannot repackage remotely from a Windows NT4 system because a file required during repackaging on that operating system, `MSCVRT.dll`, must be installed locally. Therefore, when repackaging on a Windows NT4 system, install Standalone Repackager on that machine, as described in [Installing Repackager on a Clean Machine](#).

Launching Repackager Remotely

Because you want to avoid installing applications on the clean machine, you should launch Repackager remotely from the clean machine.

To launch Repackager remotely, perform the following tasks:

- [Sharing Directories on a Machine with an Installation of AdminStudio](#)
- [Creating a Shortcut to Repackager on the Clean Machine](#)
- [Launching Remote Repackager on the Clean Machine](#)

Sharing Directories on a Machine with an Installation of AdminStudio

To share directories on a machine where AdminStudio is installed, perform the following steps:



Task:

To share the Repackager and AdminStudio Shared folders:

1. Locate a production machine with network access that has AdminStudio installed on it.



Tip • Check to make sure that this installation of AdminStudio has already been activated before proceeding.

2. Open Windows Explorer and locate the following directory:

C:\Program Files\AdminStudio\9.5\Repackager

3. Right-click the Repackager directory and then click **Sharing and Security**. The **Sharing** tab of the **Repackager Properties** dialog box opens.
4. Select **Share this folder** and configure sharing rights as necessary.
5. Click **OK** to close the dialog box.
6. Repeat the steps above to also share the AdminStudio Shared directory used by that installation of AdminStudio.

Creating a Shortcut to Repackager on the Clean Machine

To create a shortcut to Repackager on the clean machine, perform the following steps:



Task:

To create a shortcut to Repackager on the clean machine:

1. On this clean machine with network access, launch Windows Explorer.
2. Select **Map Network Drive** from the **Tools** menu. The **Map Network Drive** dialog box opens.
3. Specify the drive letter you want use to represent the shared location.
4. Click **Browse**. The **Browse for Folder** dialog box opens.
5. Select the shared Repackager directory on the production machine (that you configured in [Sharing Directories on a Machine with an Installation of AdminStudio](#)) and click OK.
6. Click **Finish** to exit the **Map Network Drive** dialog box.
7. From Windows Explorer, navigate to the drive mapped to the shared Repackager directory on the production machine.
8. Right-click on the `isl1c.exe` file (the Repackager executable file), point to **Send To**, and click **Desktop (create shortcut)**. A shortcut to Repackager in the shared directory is now on the Desktop.

Launching Remote Repackager on the Clean Machine

To launch Repackager on the clean machine, perform the following steps:



Task:

To launch Remote Repackager on the clean machine:

1. On the clean machine, double-click the **Repackager** shortcut on the desktop (that you created in [Creating a Shortcut to Repackager on the Clean Machine](#)). The Repackager Start Page opens.



Important • Because you are running Repackager remotely, the online help topics cannot be viewed. However, you can view a version of AdminStudio Help Library online at:

<http://helpnet.flexerasoftware.com>

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Configuring Repackager to Ensure Optimal Installation Capture

2. Click the **Capture an Installation Using Repackaging Wizard** link. The **Welcome** panel of the **Repackaging Wizard** opens.
3. Continue using the Repackaging Wizard to capture a legacy setup, following the instructions in [Repackaging Legacy Installations Using the Repackaging Wizard](#).



Caution • On the **Set Target Project Information and Capture Settings** panel of the Repackaging Wizard, do not set the **Project path to store files** field to a location on the clean machine; instead choose a network location.

Installing Repackager on a Clean Machine

It is essential that you repackage applications on a “clean” system to ensure you capture all changes made by the installation. A clean system typically consists of a computer with only the operating system and necessary service packs installed on it. It is the baseline system that the computer requires to run.

While you want to avoid installing applications on the clean machine, if the clean machine does not have network connectivity to an installation of Repackager (which is required in order to run Repackager remotely), you have to install Repackager locally on a clean machine by running the Repackager installation.

To install a standalone version of Repackager on a clean machine, perform the following steps.



Task: *To install Repackager on a clean machine:*

1. Build a “clean machine”—a computer with only the operating system and necessary service packs installed on it.
2. On the AdminStudio installation DVD, locate the standalone Repackager installation in the Tools directory:
`\Tools\StandaloneRepackager.exe`
3. Copy `StandaloneRepackager.exe` to the clean machine.
4. Launch the setup. The **Welcome Panel** opens.
5. Click **Next**. The **License Agreement** panel opens.
6. Select the **I accept the terms of the license agreement** option and click **Next**. The **Customer Information** panel opens.
7. Enter a **User Name** and **Organization** name to identify this installation of Repackager.
8. Enter the **Serial Number** you received for the Edition of AdminStudio that you purchased.
9. Click **Next**. The **Destination Folder** panel opens.
10. If you want to install Repackager in the specified directory, click **Next**. If you want to select a different directory, click **Change**, select a new directory, and then click **Next**. The **AdminStudio Shared Location** panel opens.
11. Specify the location of your organization’s AdminStudio Shared directory, and click **Next**. The **Ready to Install** panel opens.
12. Click **Install** to begin the installation process. The **Installing Repackager** panel opens. When installation is complete, the **InstallShield Wizard Completed** panel opens.
13. Click **Finish** to exit the Wizard. A Repackager shortcut will be added to the Windows **Start** menu under **AdminStudio, AdminStudio Tools**.

Repackaging Legacy Installations Using the Repackaging Wizard

One frequently used method of creating a Repackager project is to repackage a legacy setup. Fundamentally, this involves monitoring the execution of a non-Windows Installer setup and converting changes made by the setup into a Windows Installer file.

Repackager provides the [Repackaging Wizard](#) for accomplishing this task. Using this Wizard, you can select the repackaging method (either Snapshot or Installation Monitoring), specify the setup(s) you want to repackage, and run the setup(s). When the Repackaging Wizard has finished its analysis, Repackager automatically creates a Repackager project (.irp) file, which can be modified in Repackager. You can then convert this file to an InstallShield Editor project (.ism) for further editing, or convert it directly to a Windows Installer package (.msi).

For InstallShield Editor or DevStudio 9.x InstallScript installations, you can choose to use the [InstallShield Professional Logging Method](#) instead of the Snapshot or Installation Monitoring methods.



Caution • It is highly recommended that you repackage applications on a “clean” system. See [Configuring Repackager to Ensure Optimal Installation Capture](#) for more information.

When using the Repackaging Wizard to repackage a legacy setup, you can use any of the following methods:

Table 7-7 • Repackaging Methods


Repackaging Method	Description
Installation Monitoring Method	<p>Repackager monitors system changes as an application is installed, and that data can be converted into a Windows Installer package. Installation Monitoring is the default method.</p> <p>See Repackaging Using the Installation Monitoring Method.</p>  <hr/> <p>Edition •</p>
Snapshot Method	<p>Repackager compares a system snapshot before and after an installation, determines the changes that were made, and that data can be converted to a Windows Installer package. This is the default method.</p> <p>See Repackaging Using the Snapshot Method.</p>
InstallShield Professional Logging Method	<p>Repackager reads logged output of InstallShield Editor and DevStudio 9.x InstallScript installations, allowing you to get additional information that would not be captured by standard repackaging.</p> <p>See Repackaging Using the InstallShield Professional Logging Method.</p>

Table 7-7 • Repackaging Methods (cont.)

Repackaging Method	Description
Using InstallScript Scan	<p>You can use the Repackaging Wizard and InstallScript Scan to convert an InstallScript MSI installation to a Basic MSI package with InstallScript support. InstallScript Scan preserves the original components and much of the InstallScript installation logic, architecture, and maintainability of the original installation package.</p> <p>See Repackaging an InstallScript MSI Setup to a Basic MSI Setup</p>

Repackaging Using the Installation Monitoring Method

When you choose the **Installation Monitoring** method of repackaging, Repackager monitors system changes as an application is installed, and then you can convert that data into a Windows Installer package. Installation Monitoring is the default method.



Caution • *It is highly recommended that you repackage applications on a “clean” system. See [Configuring Repackager to Ensure Optimal Installation Capture](#) for more information.*

To repackage an installation using the Installation Monitoring method, perform the following steps:

- [Step 1: Selecting the Repackaging Method.](#)
- [Step 2: Excluding Processes \(Optional\)](#)
- [Step 3: Collecting Product Information](#)
- [Step 4: Adding Additional Setup Programs \(Optional\)](#)
- [Step 5: Set Target Project Information](#)
- [Step 6: Set Capture Settings \(Optional\)](#)
- [Step 7: Beginning the Repackaging Process](#)

Step 1: Selecting the Repackaging Method

In this step, you launch the Repackaging Wizard and select the Installation Monitoring repackaging method.

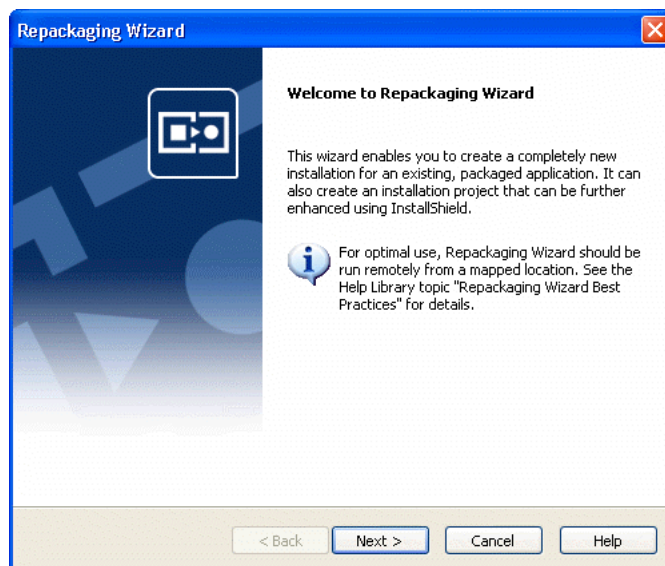
Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

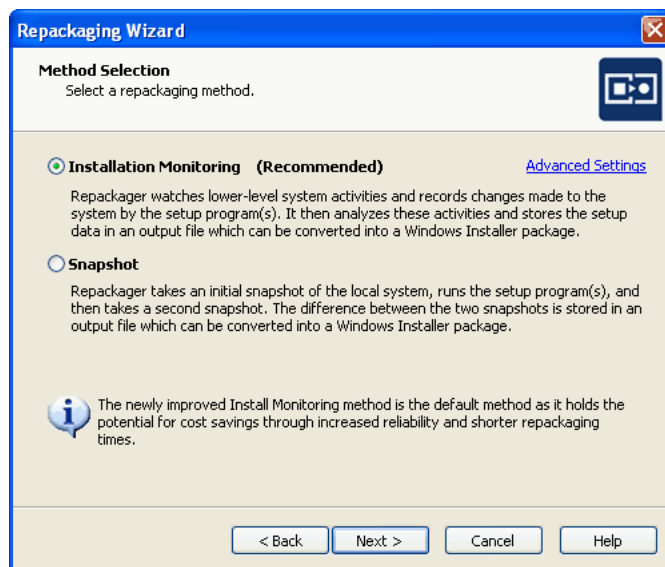


Task: *To select a repackaging method:*

1. From the Repackager interface, launch the **Repackaging Wizard** by clicking on the link or by selecting **Repackaging Wizard** from the **Tools** menu. The Welcome Panel opens.



2. Click **Next**. The Method Selection Panel opens.



3. Select **Installation Monitoring**.



Note • The InstallShield Professional Logging Method, which should be used to repackage InstallShield Editor and DevStudio 9.x InstallScript installations, is not offered as a choice on the Method Selection panel. See [Repackaging Using the InstallShield Professional Logging Method](#) for more information on using that method.

4. Continue with [Step 2: Excluding Processes \(Optional\)](#).

Step 2: Excluding Processes (Optional)

During Installation Monitoring, Repackager captures all of the activity of each service or process running on the machine, and then processes this collected data. However, many services running on a machine may have nothing to do with the installation being repackaged.

- If you want to modify the default excluded processes list, perform the following steps.
- If you do not want to modify the default excluded processes list, continue with [Step 3: Collecting Product Information](#).

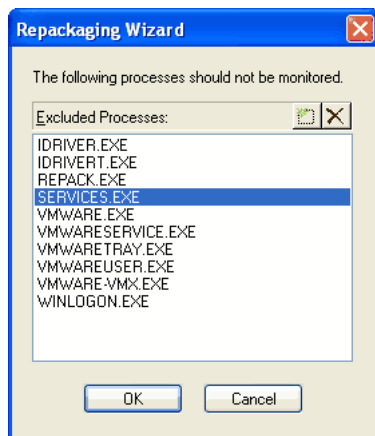




Tip • If you know that the installation that you are capturing is from a self-extracting .exe file and if you want to use the **Installation Monitoring** method, you should add the name of that .exe file to the excluded processes list.



Task: To exclude processes from Installation Monitoring:

1. On the Method Selection Panel, click the **Advanced Settings** link. The Excluded Processes dialog box opens, listing a default set of processes.



2. To add a process to this list, click the New () button to add a new blank line to this list, and enter the name of the process that you want to exclude.
3. To delete a process from this list, select the process and click the Delete () button.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard



Note • The changes you make to the excluded processes list are persisted for future Repackaging sessions. Therefore, once you have entered an appropriate set of processes to exclude for your machine, you can skip this optional step.

4. Click **OK** to return to the Method Selection Panel.
5. Continue with [Step 3: Collecting Product Information](#).


Step 3: Collecting Product Information

In this step, you will specify the installation you want to repackage and enter any command-line arguments to be used when the installation is run.



Task: To enter product information:

1. On the Method Selection Panel, click **Next**. The Collect Product Information Panel opens.

2. Click the Browse () button next to the **Program File** field and select the installation program that you are repackaging.
3. In the **Command-line Argument(s)** field, enter any command-line arguments to be used when the installation is run.
4. In the **Product Information** area, modify the **Product Name**, **Version**, and **Company Name**, as necessary.
5. If you want to associate Web sites with this installation, click the **More** link to open the Additional Product Information dialog box, enter the **Product URL** and **Support URL** for the application you are repackaging, and click OK.

6. Continue with [Step 4: Adding Additional Setup Programs \(Optional\)](#).

Step 4: Adding Additional Setup Programs (Optional)

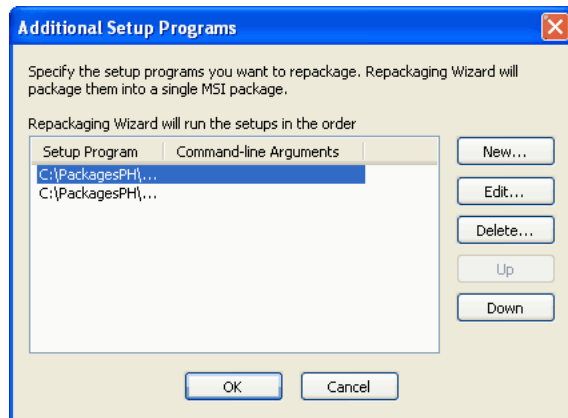
You can specify additional setup programs to repackage together with this installation. Additional setup programs share the same product name, version number, and company name in the repackaged installation. However, as you locate each additional setup program to repackage, you can specify command-line parameters pertaining only to that setup. You can also specify the order in which the setups are run, should it be necessary.

- If you want to add additional setup programs, perform the following steps.
- If you do not want to add additional setup programs, continue with [Step 5: Set Target Project Information](#).

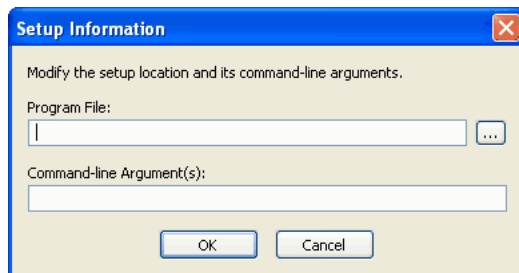


Task: To add additional setup programs, perform the following steps:

1. On the Collect Product Information Panel, click the **Edit Setup List** link. The **Additional Setup Programs** dialog box opens.



2. If you want to **add** a setup program, perform the following steps:
 - a. Click **New**. The **Setup Information** dialog box opens.

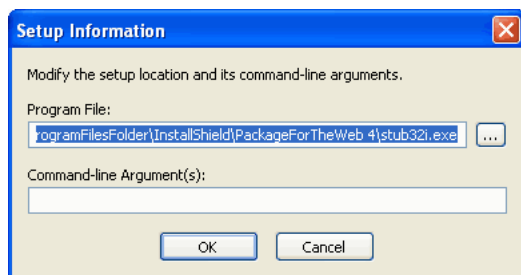


- b. Click the Browse () button next to the **Program File** field and select the setup program that you want to add.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

- c. In the **Command-line Argument(s)** field, enter any command-line arguments to be used when this setup is run.
 - d. Click **OK** to return to the **Additional Setup Programs** dialog box.
 - e. If necessary, click the **Up** and **Down** buttons to change the order in which the setups are run.
 3. If you want to **edit** an existing setup program, perform the following steps:
 - a. On the Additional Setup Programs dialog box, select the program that you want to edit and click **Edit**. The Setup Information dialog box opens.



- b. Modify the **Program File** and **Command-line Argument(s)** fields.
 - c. Click **OK** to return to the Additional Setup Programs dialog box.
 4. If you want to **delete** a listed setup program, perform the following steps:
 - a. Select the program that you want to delete and click **Delete**. A dialog box opens prompting you to confirm the deletion.
 - b. Click **OK** to confirm the deletion and return to the Additional Setup Programs dialog box, where the deleted program is no longer listed.
 5. Click **OK** to return to the **Collect Product Information** panel.
 6. Continue with [Step 5: Set Target Project Information](#).

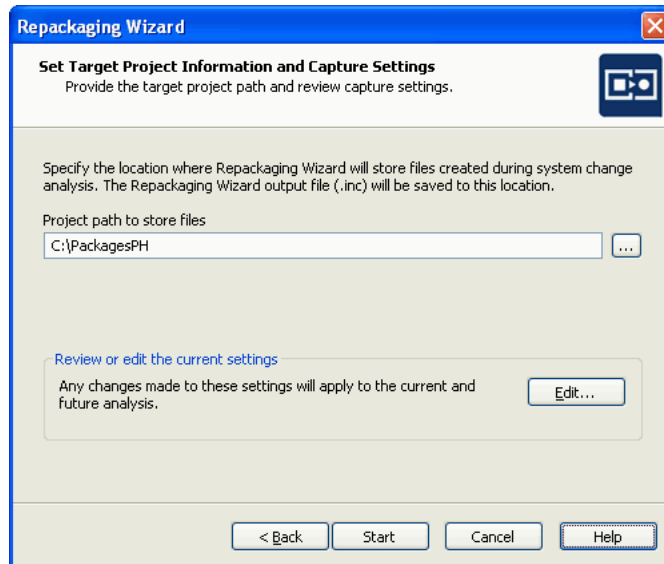
Step 5: Set Target Project Information

In this step, you identify the location where you want files created by Repackager to be stored. For the Installation Monitoring repackaging method, it is recommended that this location not be located on your clean machine, but rather on the same machine as the Repackager executable (most likely on your administrator machine).




Task: *To set target project information and capture settings:*

1. On the Collect Product Information Panel, click **Next**. The Set Target Project Information and Capture Settings Panel opens.



Note • If you specified an InstallShield Editor or DevStudio 9.x InstallScript installation in the **Program File** field of the Collect Product Information Panel, the InstallShield Professional Setup Panel appears instead of the Set Target Project Information and Capture Settings Panel. See [Repackaging Using the InstallShield Professional Logging Method](#).

2. Click the Browse () button next to the **Project path to store files** field and select the directory where you want the Repackaging Wizard to place its output, including the Repackager project file (.irp), the Repackaging Wizard output files, and source files.

You can also enter the name of a new folder in the **Project path to store files** field, and you will be prompted to create it when you exit this panel.

3. Continue with [Step 6: Set Capture Settings \(Optional\)](#).

Step 6: Set Capture Settings (Optional)

From the Set Target Project Information and Capture Settings Panel, you can specify the following capture types for the repackaging session:

- Files and deleted files
- .ini files and .ini files with non-.ini extensions
- Shortcuts

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

- Registry data and deleted registry data

Options set in this dialog box apply to the current and subsequent repackaging sessions.

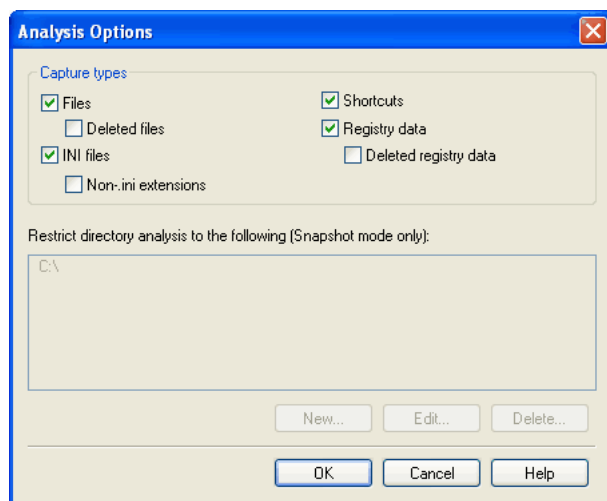
- **If you want to set capture settings**, perform the following steps.
- **If you do not want to set capture settings**, continue with [Step 7: Beginning the Repackaging Process](#).



Task:

To set capture settings:

1. On the Set Target Project Information and Capture Settings Panel, click **Edit**. The Analysis Options dialog box opens.



Note • Options set in this dialog box apply to the current and subsequent repackaging sessions.

2. Select the capture types that you want to use for this repackaging session:
 - **Files**
 - **Deleted files**
 - **INI files**
 - (INI files with) **Non-.ini extensions**
 - **Shortcuts**
 - **Registry data**
 - **Deleted registry data**
3. Click **OK** to return to the Set Target Project Information and Capture Settings Panel.
4. Continue with [Step 7: Beginning the Repackaging Process](#).

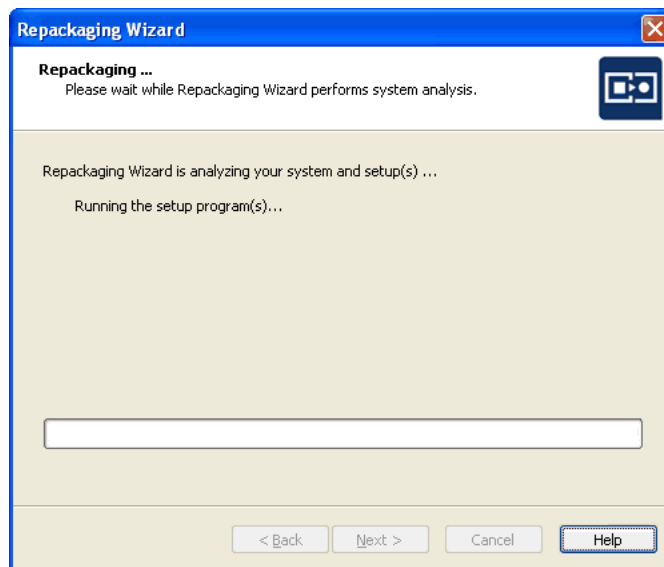
Step 7: Beginning the Repackaging Process

In this step you will begin the repackaging process.



Task: *To begin the repackaging process:*

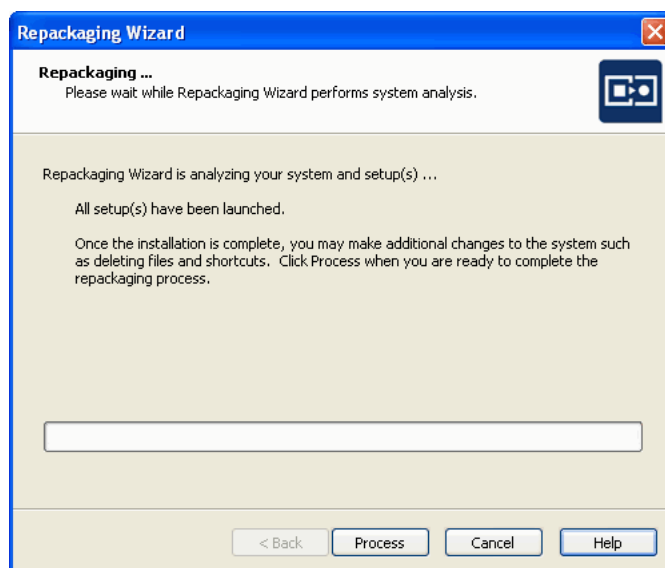
1. To begin the repackaging process, click **Start** on the Set Target Project Information and Capture Settings Panel. The **Repackaging Panel** opens and the Repackaging Wizard captures the initial system status. Then, the selected setup program will be launched.



2. Follow the prompts until the installation has completed. When the installation is complete, you are prompted to make any additional changes to the system (such as deleting files and shortcuts) that you want to be recorded in this repackaged installation.

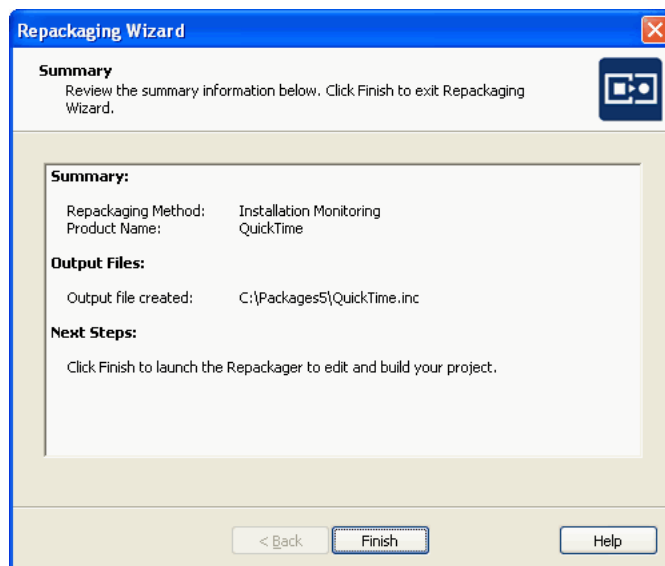
Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard



3. When you are ready to complete the repackaging process, click **Process**. The Repackaging Wizard then analyzes the system and setup data that it collected.

Following repackaging, the **Summary Panel** is displayed, providing confirmation that the repackaging was successful.



4. Click **Finish**. Repackager launches and opens the Repackager project file (*.i rp) that you just created.
5. Continue with the instructions in [Working With Repackager Projects](#).

Repackaging Using the Snapshot Method

When using the **Snapshot** method of repackaging, the Repackaging Wizard takes a reference snapshot of a system as a baseline configuration, performs the installation, and then takes a second snapshot.

The difference between the two snapshots is stored in a directory you specify, including the Repackager project file (.irp), the Repackaging Wizard output files, and the source files. The Repackager project file can then be converted into a Windows Installer package (.msi).



Caution • It is highly recommended that you repackage applications on a “clean” system. See [Configuring Repackager to Ensure Optimal Installation Capture](#) for more information.

Types of Snapshot Repackaging

There are two types of Snapshot repackaging:

Single Step

When Repackaging in a single step:

- You specify at least one setup program to repackage.
- Repackager first takes an initial system snapshot.
- Repackager then runs the setup program(s) you selected.
- Then Repackager takes a second snapshot to create the script file that can be converted into a Windows Installer package.

You also have the option of requiring the Repackager to prompt you before running the setup program(s), allowing you the opportunity to make changes to your system that you want included in the final package.

See [Performing Single Step Snapshot Repackaging](#).

Multiple Step

When repackaging in multiple steps:

- You run the Repackager to obtain an initial system snapshot, after which the Repackager exits.
- You can then perform any modifications to the system, such as changing configurations, running installations, and so forth.
- After making the necessary modifications, you would then run the Repackager again to analyze system status changes.
- Repackager compares the final snapshot to the initial snapshot to determine the system changes that were made, and then records that information in a script file.

See [Performing Multiple Step Snapshot Repackaging](#).

Performing Multiple Step Snapshot Repackaging

To repackage an installation using the **Multiple Step Snapshot** method, perform the following steps:

- [Step 1: Selecting the Repackaging Method](#).
- [Step 2: Initial Analysis](#)
- [Step 3: Install Setup and Make Manual System Changes](#)
- [Step 4: Entering Product Information](#)
- [Step 5: Set Target Project Information](#)
- [Step 6: Set Capture Settings \(Optional\)](#)
- [Step 7: Beginning the Repackaging Process](#)

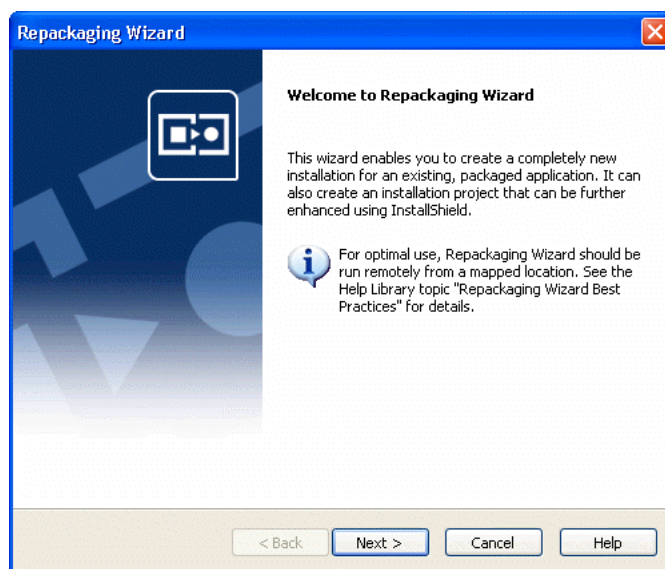
Step 1: Selecting the Repackaging Method

In this step, you launch the Repackaging Wizard and select the **Snapshot** repackaging method.

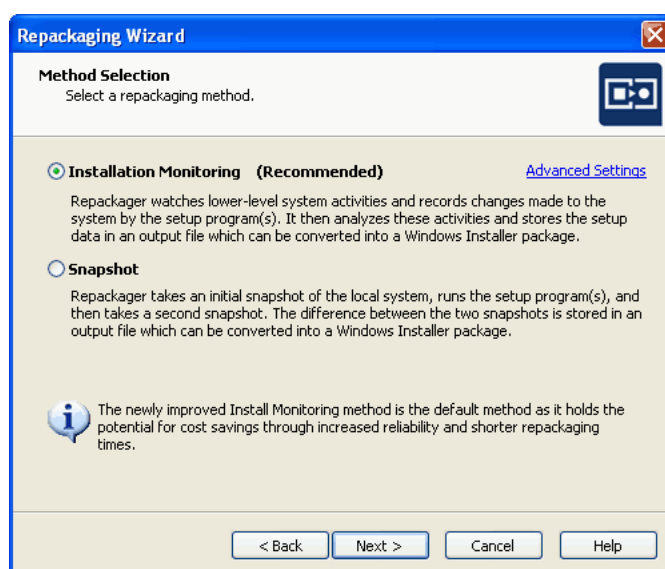


Task: *To select a repackaging method:*

1. From the Repackager interface, launch the **Repackaging Wizard** by clicking on the link or by selecting **Repackaging Wizard** from the **Tools** menu. The Welcome Panel opens.

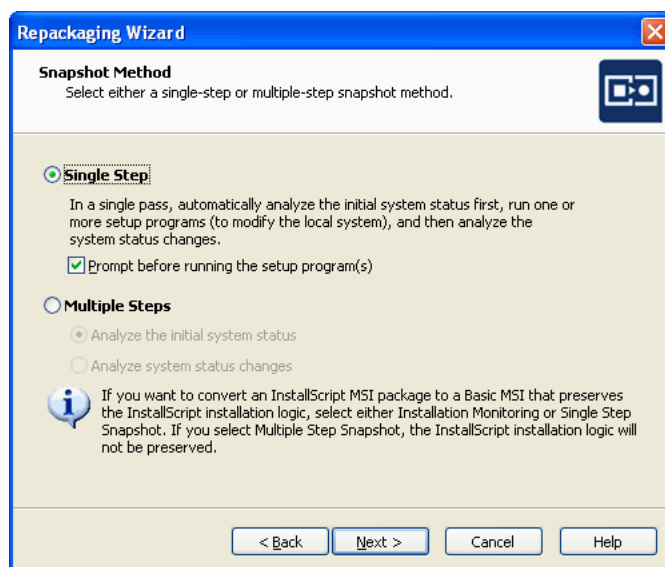


2. Click **Next**. The Method Selection Panel opens.



Note • The InstallShield Professional Logging Method, which should be used to repackage InstallShield Editor and DevStudio 9.x InstallScript installations, is not offered as a choice on the Method Selection panel. See [Repackaging Using the InstallShield Professional Logging Method](#) for more information on using that method.

3. Select **Snapshot** and click Next. The Snapshot Method panel opens.



4. On the Snapshot Method panel, select **Multiple Steps**. The **Analyze the initial system status** option is enabled.
5. Select the **Analyze the initial system status** option.
6. Continue with [Step 2: Initial Analysis](#).

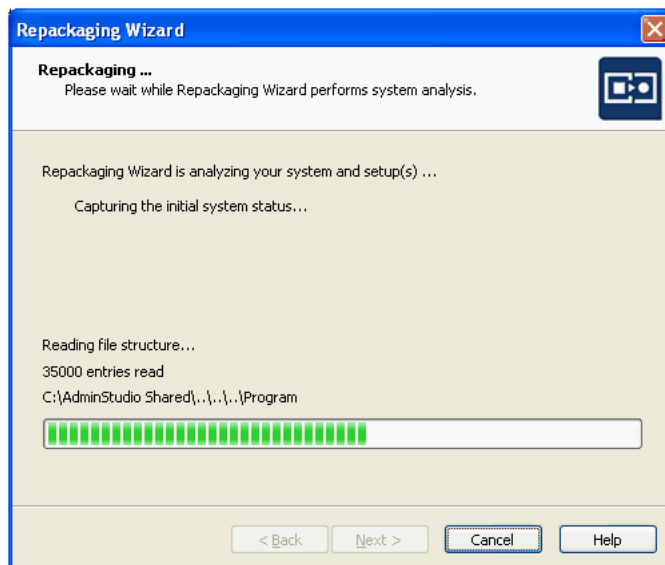
Step 2: Initial Analysis

In this step, the Repackaging Wizard takes an initial snapshot of your system.

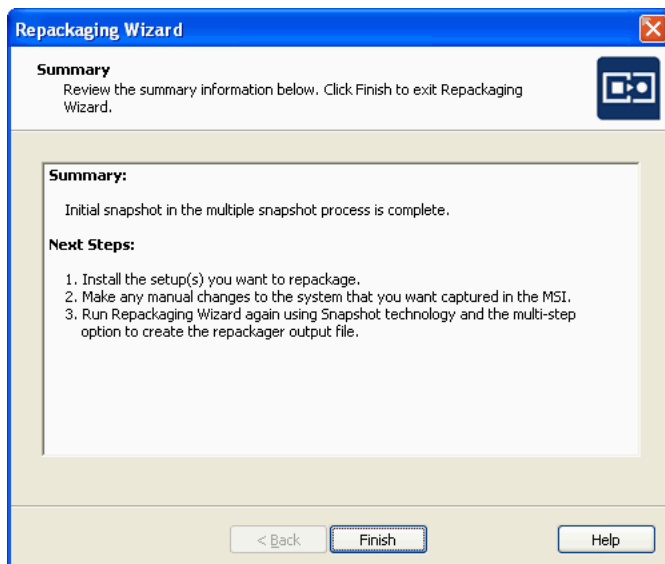


Task: *To perform initial analysis:*

1. On the Snapshot Method panel, click **Next**. The Repackaging Panel of the Repackaging Wizard opens, displaying the progress of the initial system status capture.



When Repackager finishes capturing the initial system status, the **Summary** panel opens, prompting you to install the application you are repackaging.



2. Click **Finish** to close the Repackaging Wizard.
3. Continue with [Step 3: Install Setup and Make Manual System Changes](#).

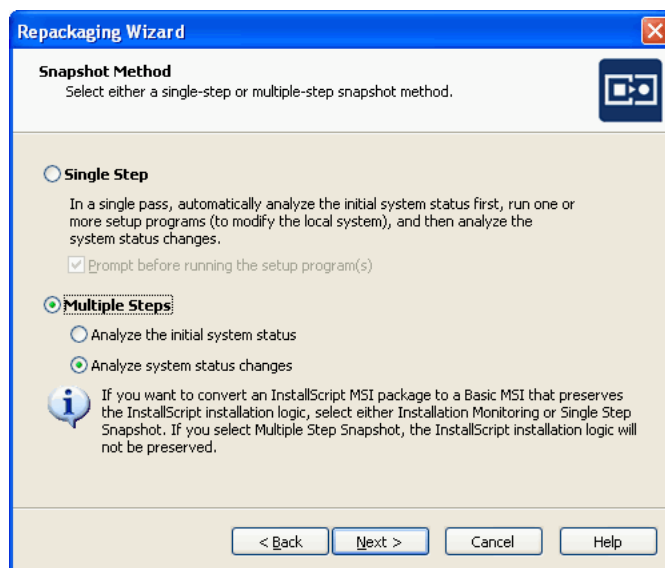
Step 3: Install Setup and Make Manual System Changes

In this step, you will manually launch the installation of the application you are repackaging, and then you will make any manual changes to the system that you want captured in the Windows Installer package.



Task: *To install setup and make manual system changes:*

1. Launch the installation program of the application you are repackaging.
2. Follow the prompts until the installation has completed.
3. When the installation is complete, make any additional changes to the system (such as deleting files and shortcuts) that you want to be recorded in this repackaged installation.
4. Launch the Repackaging Wizard again. The **Welcome Panel** opens.
5. Click **Next**. The **Method Selection Panel** opens.
6. Select **Snapshot** and click **Next**. The **Snapshot Method Panel** opens with **Multiple Steps** already selected, and the **Analyze system status changes** option now enabled and selected.



7. Continue with [Step 4: Entering Product Information](#).

Step 4: Entering Product Information

In this step, you will enter product information for the application that you just installed.



Task: *To enter product information:*

1. On the **Snapshot Method Panel**, click **Next**. The **Collect Product Information Panel** opens. Because you are now performing the second step of a multiple-step Snapshot, the **Setup Programs** area is disabled (because you have already installed the application you are repackaging).

The screenshot shows the 'Repackaging Wizard' dialog box with the 'Collect Product Information' tab selected. The dialog has a title bar with a close button. Below the title bar, it says 'Collect Product Information' and 'Provide product information for repackaging. Items with asterisks (*) are required.' There are two main sections: 'Setup Programs' and 'Product Information'. The 'Setup Programs' section is disabled and contains a checkbox, a 'Program File' field with a browse button, and a 'Command-line Argument(s)' field. The 'Product Information' section is active and contains 'Product Name', 'Version', and 'Company Name' fields, each with an asterisk indicating it is required. There is a 'More' link next to the 'Company Name' field. At the bottom, there are four buttons: '< Back', 'Next >', 'Cancel', and 'Help'.

2. In the **Product Information** area, modify the **Product Name**, **Version**, and **Company Name**, as necessary.
3. If you want to associate Web sites with this installation package, perform the following steps:
 - a. Click the **More** link. The **Additional Product Information** dialog box opens.

The screenshot shows the 'Additional Product Information' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'Product URL' and 'Support URL'. At the bottom, there are two buttons: 'OK' and 'Cancel'.

- b. Enter the **Product URL** and **Support URL** for the application you are repackaging.
 - c. Click **OK**.
4. Continue with [Step 5: Set Target Project Information](#).

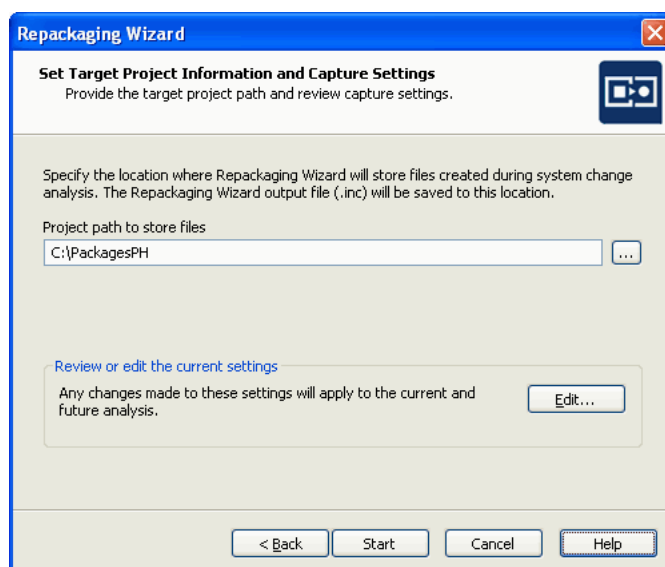
Step 5: Set Target Project Information

In this step, you identify the location where you want files created by Repackager to be stored.



Task: *To set target project information and capture settings:*

1. On the **Collect Product Information Panel**, click **Next**. The **Set Target Project Information and Capture Settings Panel** opens.



2. Click the Browse (...) button next to the **Project path to store files** field and select the directory where you want the Repackaging Wizard to place its output, including the Repackager project file (.irp), the Repackaging Wizard output files, and source files.

You can also enter the name of a new folder in the **Project path to store files** field, and you will be prompted to create it when you exit this panel.

3. Continue with [Step 6: Set Capture Settings \(Optional\)](#).

Step 6: Set Capture Settings (Optional)

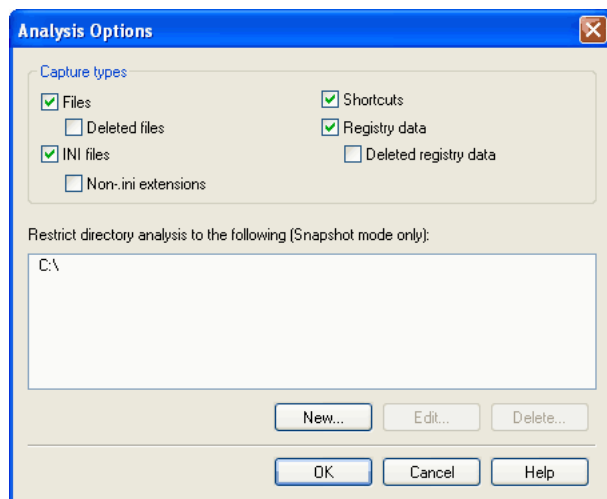
From the **Set Target Project Information and Capture Settings Panel**, you can specify capture types for the repackaging session such as files, .ini files, shortcuts, and Registry data. You can also restrict directory analysis to specific directories, which can significantly improve repackaging performance.

- **If you want to modify the default capture settings**, perform the following steps.
- **If you do not want to modify the default capture settings**, click **Next** and continue with [Step 7: Beginning the Repackaging Process](#).



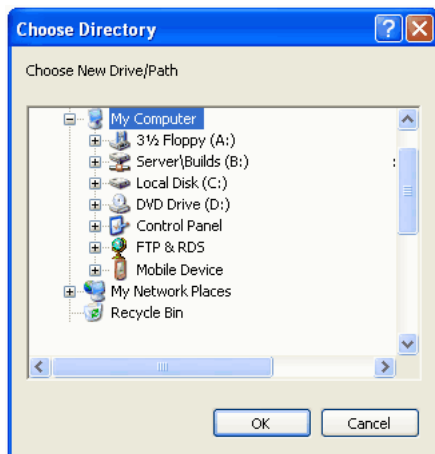
Task: *To modify capture settings:*

1. On the **Set Target Project Information and Capture Settings Panel**, click **Edit**. The **Analysis Options** dialog box opens.



Note • Options set in this dialog box apply to the current and subsequent repackaging sessions.

2. Select the capture types that you want to use for this repackaging session:
 - **Files**
 - **Deleted files**
 - **INI files**
 - (INI files with) **Non-.ini extensions**
 - **Shortcuts**
 - **Registry data**
 - **Deleted registry data**
3. If you want to restrict directory analysis to specific directories, click **New**. The **Choose Directory** dialog box opens.



4. Select a directory and click **OK**. The selected directory is now listed on the Analysis Option dialog box.
5. If you want to modify an existing restriction, or delete a restriction, select the directory and click **Edit** or **Delete**.
6. Click **OK** to return to the **Set Target Project Information and Capture Settings Panel**.
7. Continue with [Step 7: Beginning the Repackaging Process](#).

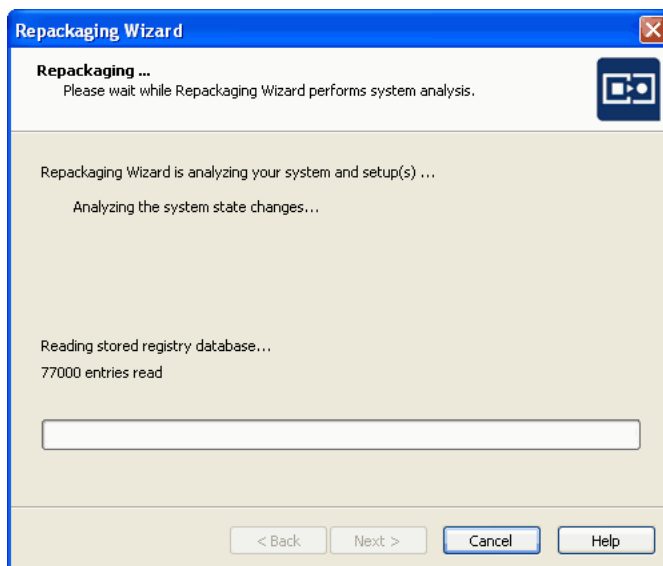
Step 7: Beginning the Repackaging Process

In this step you will begin the repackaging process.

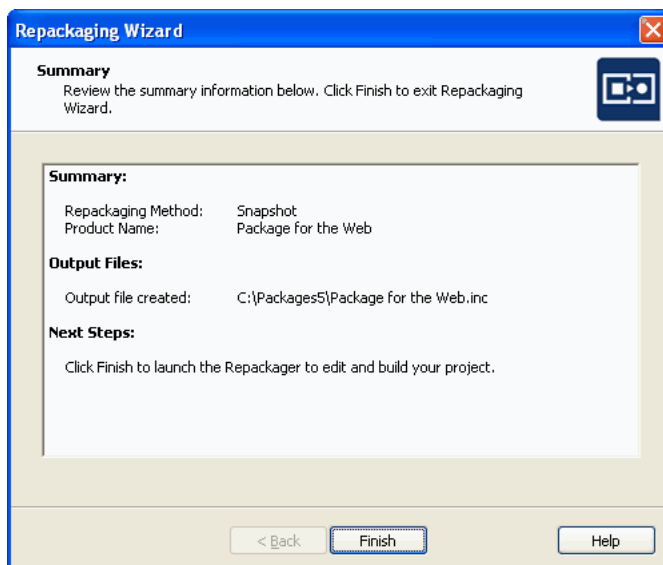


Task: *To begin the repackaging process:*

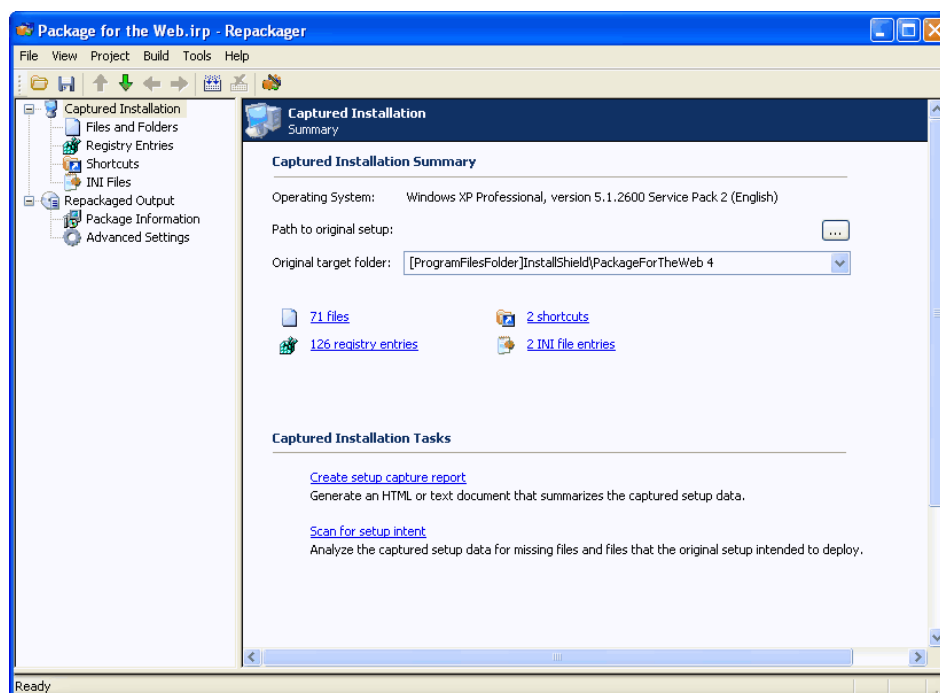
1. To begin the repackaging process, click **Start** on the Set Target Project Information and Capture Settings Panel. The **Repackaging Panel** opens and the Repackaging Wizard captures the system state changes.



When the Repackaging Wizard has finished analyzing the system state changes and creating the Repackager project, the Summary Panel opens, providing confirmation that the repackaging was successful and listing the location of your new Repackager project.



2. Click **Finish**. Repackager launches and opens the Repackager project file (*.i rp) that you just created.



3. Continue with the instructions in [Working With Repackager Projects](#).

Performing Single Step Snapshot Repackaging

To repackage an installation using the **Single Step Snapshot** method, perform the following steps:

- [Step 1: Selecting the Repackaging Method](#).
- [Step 2: Collecting Product Information](#)
- [Step 3: Set Target Project Information](#)
- [Step 4: Set Capture Settings \(Optional\)](#)
- [Step 5: Beginning the Repackaging Process](#)

Step 1: Selecting the Repackaging Method

In this step, you launch the Repackaging Wizard and select the **Snapshot** repackaging method.

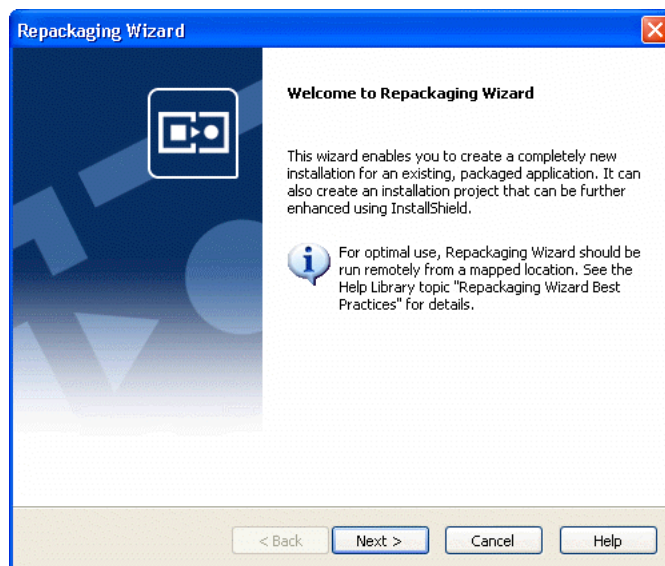
Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

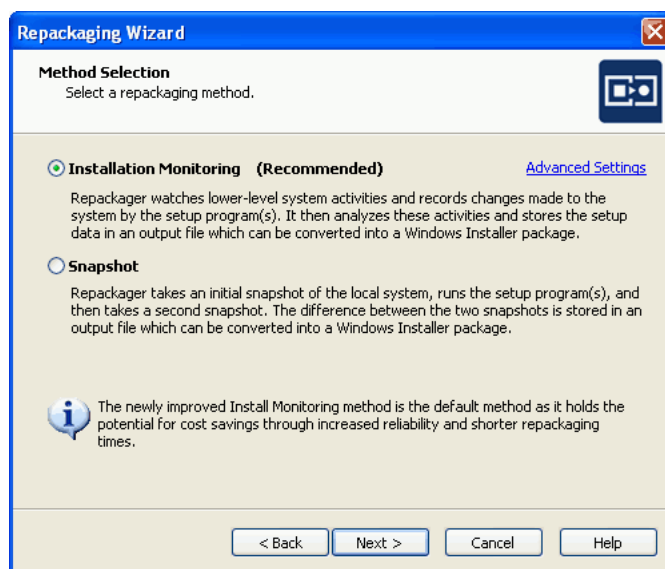


Task: To select a repackaging method:

1. From the Repackager interface, launch the **Repackaging Wizard** by clicking on the link or by selecting **Repackaging Wizard** from the **Tools** menu. The Welcome Panel opens.

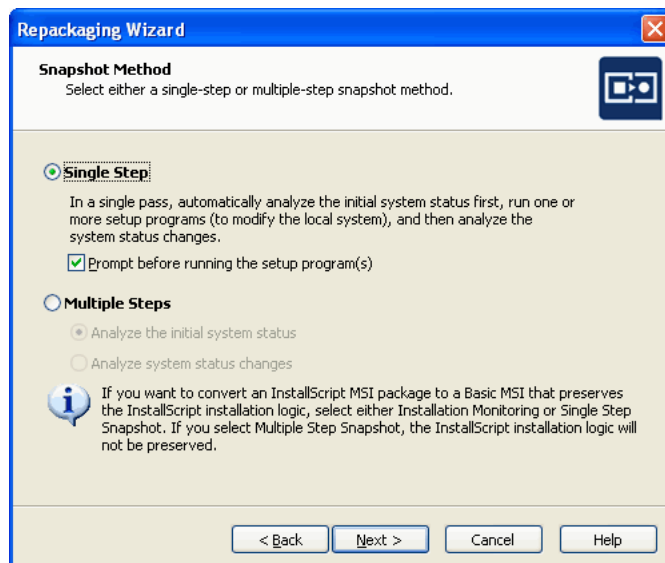


2. Click **Next**. The Method Selection Panel opens.



Note • The InstallShield Professional Logging Method, which should be used to repackage InstallShield Editor and DevStudio 9.x InstallScript setups, is not offered as a choice on the Method Selection panel. See [Repackaging Using the InstallShield Professional Logging Method](#) for more information on using that method.

3. Select **Snapshot** and click Next. The Snapshot Method panel opens.



4. On the Snapshot Method panel, select **Single Step**.
5. If you want to be prompted before the selected setup program is launched, select the **Prompt before running the setup program(s) option**. If you do not select this option, the setup program will automatically be launched as soon as the Repackaging Wizard has finished analyzing the system status.
6. Continue with [Step 2: Collecting Product Information](#).

Step 2: Collecting Product Information

In this step, you will specify the installation you want to repackage and enter any command-line arguments to be used when the installation is run.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard


Repackaging Legacy Installations Using the Repackaging Wizard



Task: To enter product information:

1. On the Snapshot Method panel, click **Next**. The Collect Product Information Panel opens with the **Setup Programs** and **Product Information** areas enabled.

The screenshot shows the 'Repackaging Wizard' dialog box with the 'Collect Product Information' tab selected. The dialog has a title bar with a close button. Below the title bar, it says 'Collect Product Information' and 'Provide product information for repackaging. Items with asterisks (*) are required.' There are two main sections: 'Setup Programs' and 'Product Information'. The 'Setup Programs' section has a 'Program File:' field with a browse button (...), a 'Command-line Argument(s):' field, and an 'Edit Setup List' link. The 'Product Information' section has 'Product Name:', 'Version:', and 'Company Name:' fields. The 'Version:' field has '1.0' entered. There is a 'More' link in the bottom right of the 'Product Information' section. At the bottom of the dialog are buttons for '< Back', 'Next >', 'Cancel', and 'Help'.

2. Click the Browse () button next to the **Program File** field and select the installation program that you are repackaging.
3. In the **Command-line Argument(s)** field, enter any command-line arguments to be used when the installation is run.
4. In the **Product Information** area, modify the **Product Name**, **Version**, and **Company Name**, as necessary.
5. If you want to associate Web sites with this installation package, perform the following steps:
 - a. Click the **More** link. The **Additional Product Information** dialog box opens.

The screenshot shows the 'Additional Product Information' dialog box. It has a title bar with a close button. Inside, there are two text input fields: 'Product URL:' and 'Support URL:'. At the bottom are 'OK' and 'Cancel' buttons.

- b. Enter the **Product URL** and **Support URL** for the application you are repackaging.
 - c. Click **OK**.
6. Continue with [Step 3: Set Target Project Information](#).

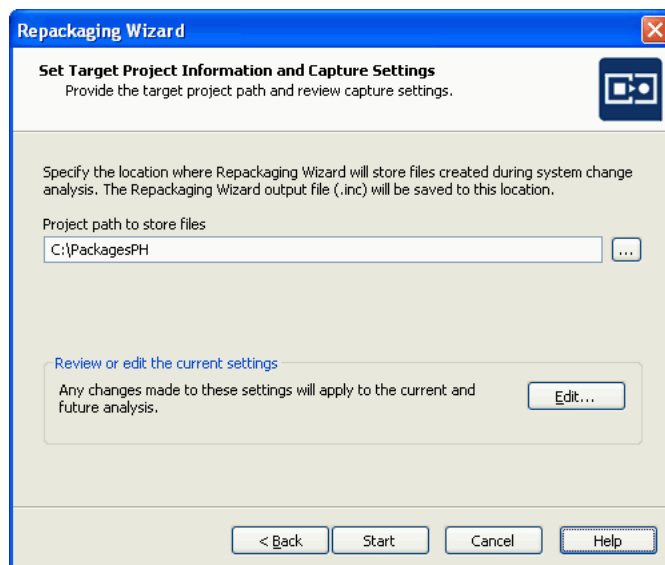
Step 3: Set Target Project Information

In this step, you identify the location where you want files created by Repackager to be stored.



Task: *To set target project information:*

1. On the Collect Product Information Panel, click **Next**. The **Set Target Project Information and Capture Settings Panel** opens.



2. Click the Browse (...) button next to the **Project path to store files** field and select the directory where you want the Repackaging Wizard to place its output, including the Repackager project file (.irp), the Repackaging Wizard output files, and source files.

You can also enter the name of a new folder in the **Project path to store files** field, and you will be prompted to create it when you exit this panel.

3. Continue with [Step 4: Set Capture Settings \(Optional\)](#).

Step 4: Set Capture Settings (Optional)

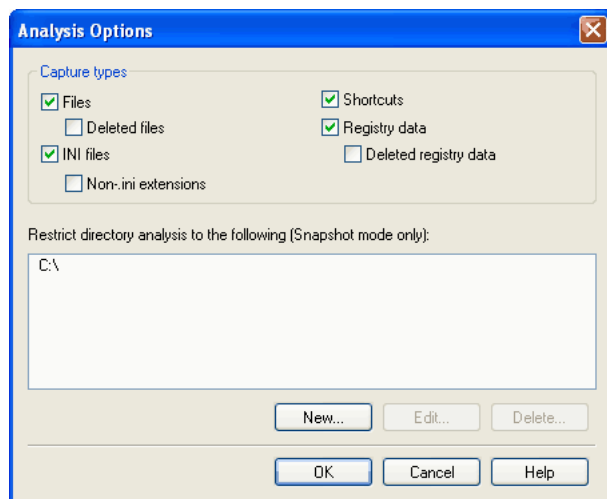
From the **Set Target Project Information and Capture Settings Panel**, you can specify capture types for the repackaging session such as files, .ini files, shortcuts, and Registry data. You can also restrict directory analysis to specific directories, which can significantly improve repackaging performance.

- **If you want to modify the default capture settings**, perform the following steps.
- **If you do not want to modify the default capture settings**, click **Next** and continue with [Step 7: Beginning the Repackaging Process](#).



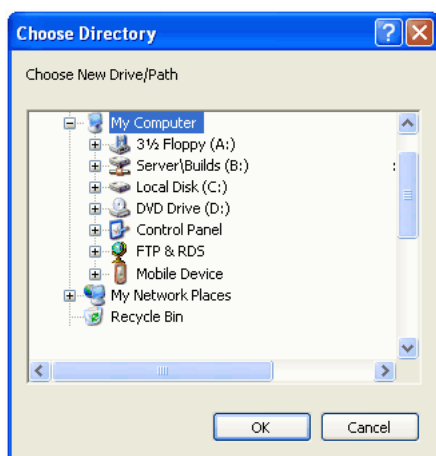
Task: *To modify capture settings:*

1. On the **Set Target Project Information and Capture Settings Panel**, click **Edit**. The **Analysis Options** dialog box opens.



Note • Options set in this dialog box apply to the current and subsequent repackaging sessions.

2. Select the capture types that you want to use for this repackaging session:
 - **Files**
 - **Deleted files**
 - **INI files**
 - (INI files with) **Non-.ini extensions**
 - **Shortcuts**
 - **Registry data**
 - **Deleted registry data**
3. If you want to restrict directory analysis to specific directories, click **New**. The **Choose Directory** dialog box opens.



4. Select a directory and click **OK**. The selected directory is now listed on the Analysis Option dialog box.
5. If you want to modify an existing restriction, or delete a restriction, select the directory and click **Edit** or **Delete**.
6. Click **OK** to return to the **Set Target Project Information and Capture Settings Panel**.
7. Continue with [Step 7: Beginning the Repackaging Process](#).

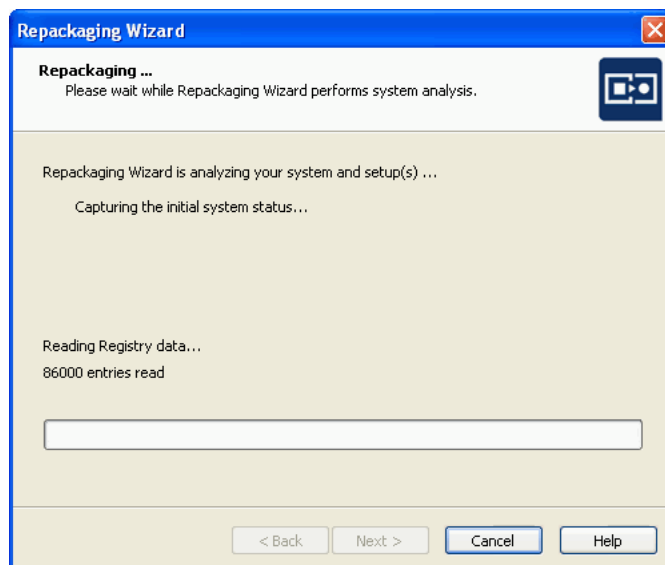
Step 5: Beginning the Repackaging Process

In this step you will begin the repackaging process.

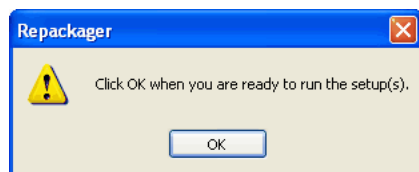


Task: *To begin the repackaging process:*

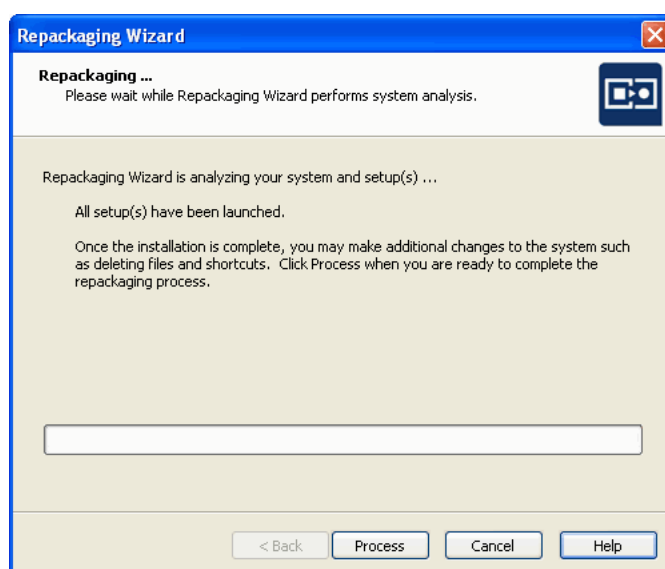
1. To begin the repackaging process, click **Start** on the **Set Target Project Information and Capture Settings Panel**. The **Repackaging Panel** opens and the Repackaging Wizard captures the initial system status.



Depending upon whether you chose the **Prompt before running the setup program(s)** option on the Snapshot Method Panel, either the installation that you selected will start or you will be prompted to start it.

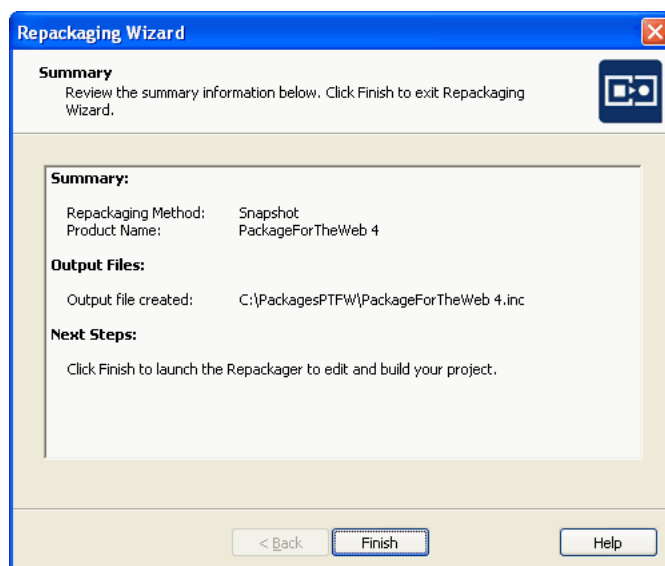


2. Install the application by following the prompts until the installation has completed.
3. When the installation is complete, you are prompted to make any additional changes to the system (such as deleting files and shortcuts) that you want to be recorded in this repackaged installation.



4. When you are ready to complete the repackaging process, click **Process**. The Repackaging Wizard then analyzes the system and setup data that it collected.

Following repackaging, the **Summary Panel** is displayed, providing confirmation that the repackaging was successful and listing the location of the Repackager project that was just created.



5. Click **Finish**. Repackager launches and opens the Repackager project file (*.i rp) that you just created.
6. Continue with the instructions in [Working With Repackager Projects](#).

Repackaging Using the InstallShield Professional Logging Method

If you are repackaging an InstallShield Editor or DevStudio 9.x InstallScript installation, you can use the InstallShield Professional Logging Method to read logged output of those setups to obtain additional information that would not be captured by the Installation Monitoring or Snapshot methods: [Path Variables](#) and the [Feature Tree](#).

See [InstallShield Professional Logging Method](#) for more information.

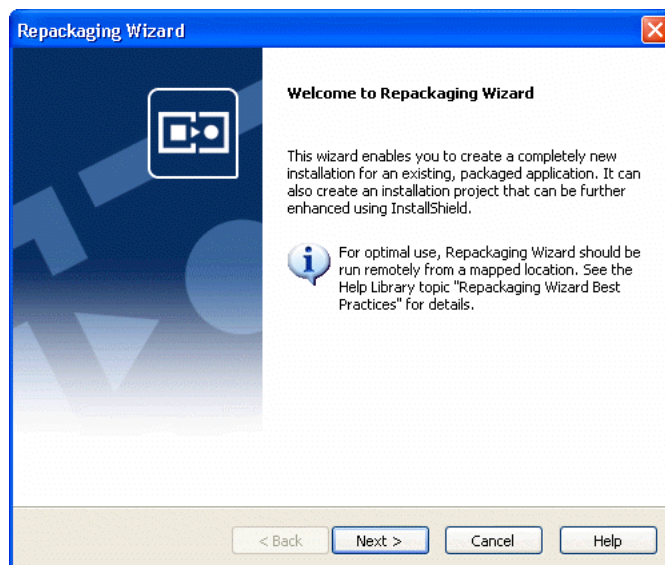
Using the InstallShield Professional Logging Method

To repackage an InstallShield Editor or DevStudio 9.x InstallScript installation using the InstallShield Professional Logging Method, perform the following steps:

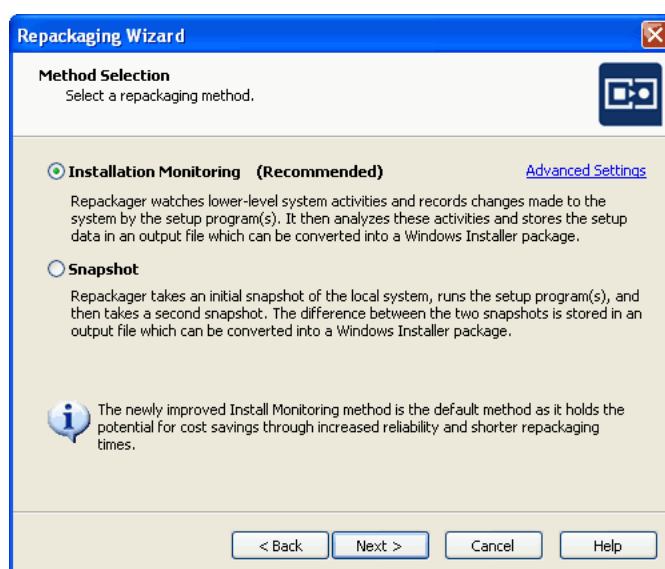


Task: *To repackage using the InstallShield Professional Logging Method:*

1. Launch the **Repackaging Wizard** from Repackager. The **Welcome Panel** opens.



2. Click **Next**. The Method Selection Panel opens.



3. Select the **Installation Monitoring** method.

The **InstallShield Professional Logging Method** is not offered as a choice on this panel, but when you select an InstallShield Editor or DevStudio 9.x InstallScript installation on the **Collect Product Information Panel** (the next panel in the Repackaging Wizard), Repackager will automatically detect the InstallScript installation and will display the **InstallShield Professional Setup Panel**. On this panel you can choose to use the **InstallShield Professional Logging Method** instead of the method you chose on the Method Selection Panel.



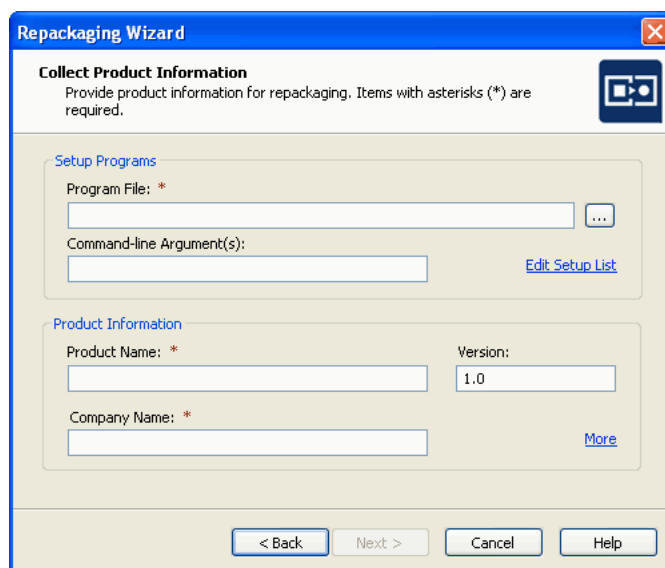
Caution • When you are attempting to repackage a setup using the InstallShield Professional Logging Method, you are permitted to select either **Installation Monitoring** or **Snapshot** on the Method Selection Panel, but it is recommended that you select **Installation Monitoring**.

If you select the **Snapshot** method, an additional panel opens, entitled *Snapshot Method*. On this panel, if you select **Multiple Steps** instead of **Single Step**, Repackager will not recognize the setup as an InstallShield Editor or DevStudio 9.x InstallScript installation and the **InstallShield Professional Setup Panel** will not open. You can avoid this extra and possibly troublesome step by selecting **Installation Monitoring** on the Method Selection Panel.

4. On the **Method Selection Panel**, click **Next**. The **Collect Product Information Panel** opens.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard



The Repackaging Wizard dialog box is titled "Repackaging Wizard" and has a close button (X) in the top right corner. It contains two main sections: "Collect Product Information" and "Setup Programs".

Collect Product Information
Provide product information for repackaging. Items with asterisks (*) are required.

Setup Programs
Program File: * (text field with a browse button "...")
Command-line Argument(s): (text field) [Edit Setup List](#)

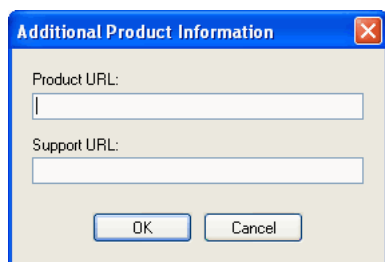
Product Information
Product Name: * (text field) Version: (text field with "1.0")
Company Name: * (text field) [More](#)

Navigation buttons at the bottom: < Back, Next >, Cancel, Help.



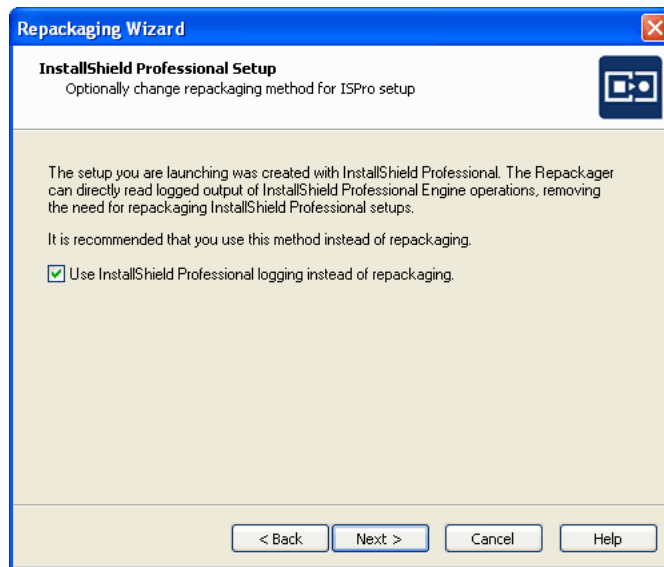
Caution • Do not use the **Edit Setup List** link to add an additional setup program to this project. If you do, Repackager will not recognize the specified installation as an InstallShield Editor or DevStudio 9.x InstallScript installation and the **InstallShield Professional Setup Panel** will not open.

5. Click the Browse (...) button next to the **Program File** field and select the InstallShield Editor or DevStudio 9.x InstallScript installation that you are repackaging.
6. In the **Command-line Argument(s)** field, enter any command-line arguments to be used when the installation is run.
7. In the **Product Information** area, modify the **Product Name**, **Version**, and **Company Name**, as necessary.
8. If you want to associate Web sites with this installation package, perform the following steps:
 - a. Click the **More** link. The **Additional Product Information** dialog box opens.

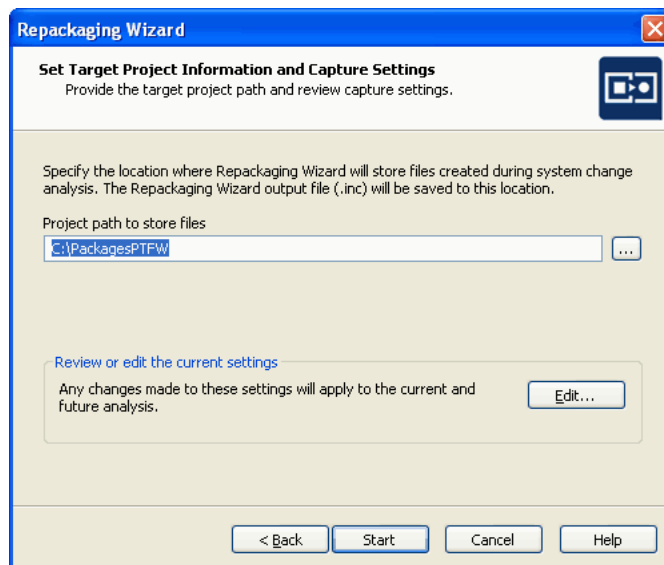


The Additional Product Information dialog box is titled "Additional Product Information" and has a close button (X) in the top right corner. It contains two text fields: "Product URL:" and "Support URL:". At the bottom are "OK" and "Cancel" buttons.

- b. Enter the **Product URL** and **Support URL** for the application you are repackaging.
 - c. Click **OK**.
9. Click **Next**. Because you specified an InstallShield Editor or DevStudio 9.x InstallScript installation in the **Program File** field, the InstallShield Professional Setup Panel opens.



10. Select the **Use InstallShield Professional logging instead of repackaging** option.
11. Click **Next**. The **Set Target Project Information and Capture Settings Panel** opens.



12. Click the Browse (...) button next to the **Project path to store files** field and select the directory where you want the Repackaging Wizard to place its output, including the Repackager project file (.irp), the Repackaging Wizard output files, and source files.

You can also enter the name of a new folder in the **Project path to store files** field, and you will be prompted to create it when you exit this panel.

13. To begin the repackaging process, click **Start** on the Set Target Project Information and Capture Settings Panel. The selected InstallShield Editor or DevStudio 9.x InstallScript installation is launched.
14. Follow the prompts until the installation has completed.

Repackager then launches and opens the Repackager project file (*.irp) that you just created.

15. Continue with the instructions in [Working With Repackager Projects](#).

Repackaging an InstallScript MSI Setup to a Basic MSI Setup

InstallScript MSI installations use a Windows Installer database for storage of all file/registry information, but the actual user interface, and much of the installation logic is driven by the InstallScript engine via a setup.exe file. This type of installation architecture can cause difficulties during deployment, such as:

- inability to customize or transform the application
- inability to perform conflict detection
- inability to suppress the user interface
- difficulty patching or upgrading the application

Also, if an InstallScript MSI installation is repackaged using traditional methods (OS Snapshot or Installation Monitoring), significant platform-specific or custom installation, maintenance, and uninstallation logic, and user interface information is lost because those methods only record the installation activities for the specific platform used during repackaging.

Therefore, it is recommended that you use InstallScript Scan to convert an InstallScript MSI installation to a Basic MSI package with InstallScript support. InstallScript Scan preserves the original components and much of the InstallScript installation logic, architecture, and maintainability of the original installation package.



Note • If you want to convert an InstallScript MSI package to a Basic MSI package that preserves the InstallScript installation logic, and you are using the Snapshot method, you must select *Single Step* rather than *Multiple Steps*. If you select *Multiple Steps*, the InstallScript installation logic will not be preserved.



Task:

To convert an InstallScript MSI Setup to a Basic MSI Setup with InstallScript support:

1. Launch the **Repackaging Wizard** from Repackager. The **Welcome Panel** opens.
2. Click Next. The **Method Selection Panel** opens.
3. Select a **repackaging method**: Installation Monitoring or Snapshot.
4. Click Next. If you selected **Snapshot** on the **Method Selection Panel**, the **Snapshot Method Panel** appears. (If you selected **Installation Monitoring**, skip to Step 6.)
5. Select **Single Step** and click **Next**. The **Collect Product Information Panel** opens.



Caution • Because you are converting an InstallScript MSI package to a Basic MSI package with InstallScript support, you must select the *Single Step Snapshot* method (or use the *Installation Monitoring* method). If you select *Multiple Step Snapshot*, the InstallScript installation logic will not be preserved.

6. On the **Collect Product Information Panel**, select the InstallScript MSI setup file and enter other product information.



Caution • While it is possible to click the *Edit Setup List* button and select additional setups, because you are converting an InstallScript MSI package, do not select additional setups.

7. Click Next. Repackager will automatically determine if this is an InstallScript-based setup. If it is an InstallScript-based setup, the **InstallScript MSI Identified Panel** opens, informing you that the Repackaging Wizard has identified this setup as being an InstallScript MSI setup and prompting you to use InstallScript Scan to convert this setup.
8. Select Yes and click Next. The **InstallScript MSI Conversion Output Panel** opens.
9. In the Project Path to store files field, specify the location where you want the Repackaging Wizard to store files created during InstallScript Scan Analysis and where it will save the converted MSI package.



Note • To specify capture types for the repackaging session, click the *Edit* button to access the **Analysis Options** dialog box.

10. Click Next. The **Repackaging Panel** appears, displaying the progress of the repackaging operation.
11. Following repackaging, the **Summary Panel** is displayed, providing confirmation that the repackaging was successful.
12. Click Finish to launch the Repackager to edit and build your project. See [Working With Repackager Projects](#).

Running the Repackaging Wizard from the Command Line

To run the Repackaging Wizard from the command line, perform the following steps.



Task: *To run the Repackaging Wizard from the command line:*

1. Open a command-line prompt.
2. Type Repack.exe followed by any command-line options you want to pass. See [Repackaging Wizard Command-Line Options](#).
3. Press Enter.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

An example of a typical command line is as follows:

```
Repack.exe -app Setup.exe -o C:\MyRepackagedApps\Output  
-pp SomeApp -cs Custom -cf MyOptions.ini -sb
```

In the above example, the following options are used:

Table 7-8 • Repackager Command-Line Options used in Example

Option	Description
-app	Specifies the name of the setup.
-o	Specifies the location of the output directory
-pp	Specifies the name of the product (and the name of the Repackager output file).
-cs	Specifies the name of the custom analysis options file to use.
-cf	Name of the analysis options file to use.
-sb	Allows you to run Repackager silently, with no user interaction.

Repackaging a Windows Installer (.msi) Package

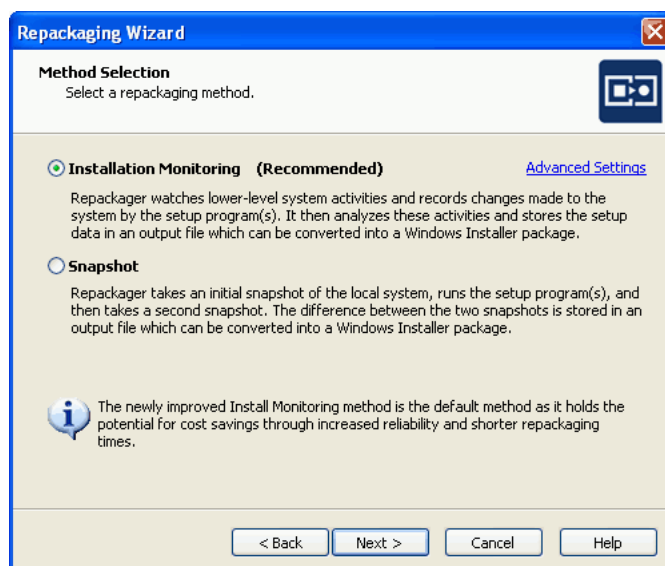
While it is not recommended that you repackage a Windows Installer (.msi) package, it sometimes may be necessary to repackage a Windows Installer package in order to convert it to a virtual package (perhaps due to the use of custom actions or other features that are not supported in application virtualization).

To repackage a Windows Installer (.msi) package, perform the following steps:



Task: *To repackage a Windows Installer package:*

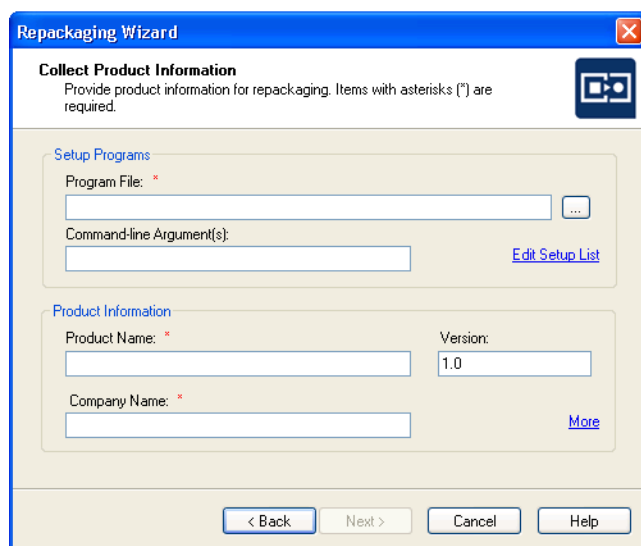
1. From the Repackager interface, launch the **Repackaging Wizard** by clicking on the link or by selecting **Repackaging Wizard** from the **Tools** menu. The Welcome Panel opens.
2. Click **Next**. The **Method Selection Panel** opens.



3. Select **Installation Monitoring** and click **Next**. The **Collect Product Information Panel** opens.



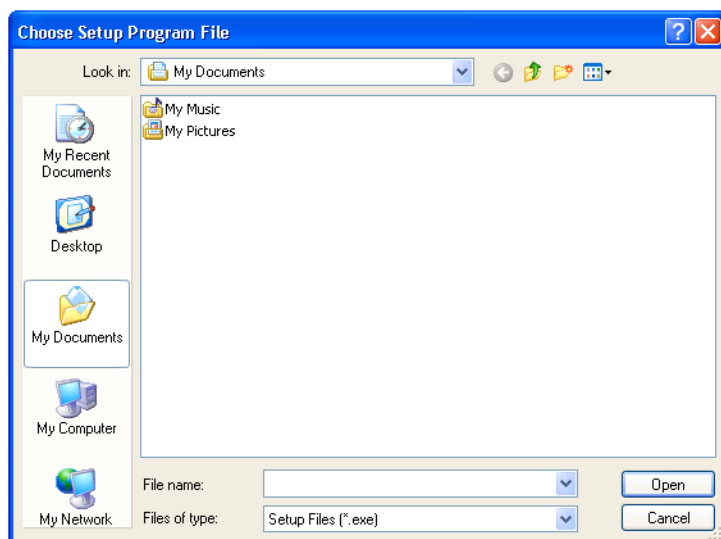
Note • The **Installation Monitoring** method is recommended, but you may also choose the **Snapshot** method when repackaging a Windows Installer package. The **Installation Monitoring** method was used in the instructions that follow. For instructions on using the **Snapshot** method, see [Repackaging Using the Snapshot Method](#)



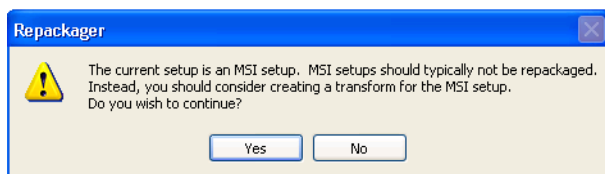
4. Click the Browse (...) button next to the **Program File** field to open the **Choose Setup Program File** dialog box.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

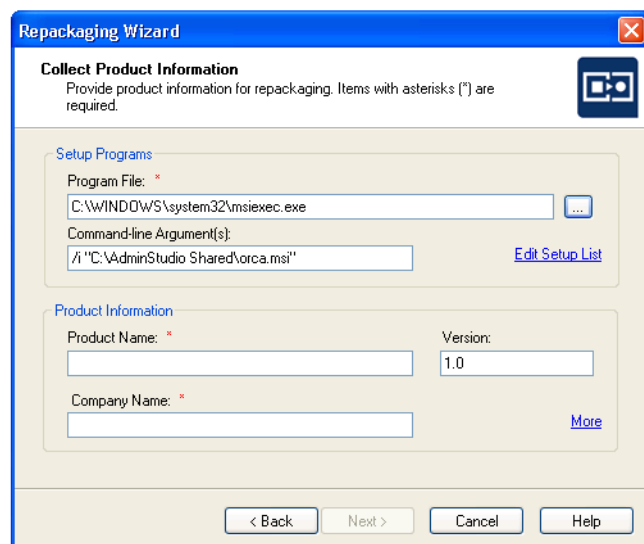
Repackaging Legacy Installations Using the Repackaging Wizard



5. From the **Files of type** list, select **All Files (*.*)**. All files in the selected directory are listed.
6. Click **Open** and select the Windows Installer package (.msi) that you are repackaging. A message appears warning you that MSI setups should not typically be repackaged.



7. Click **Yes** to close the message. Several fields in the Collection Product Information panel have been populated with the commands necessary to repackage a Windows Installer package.



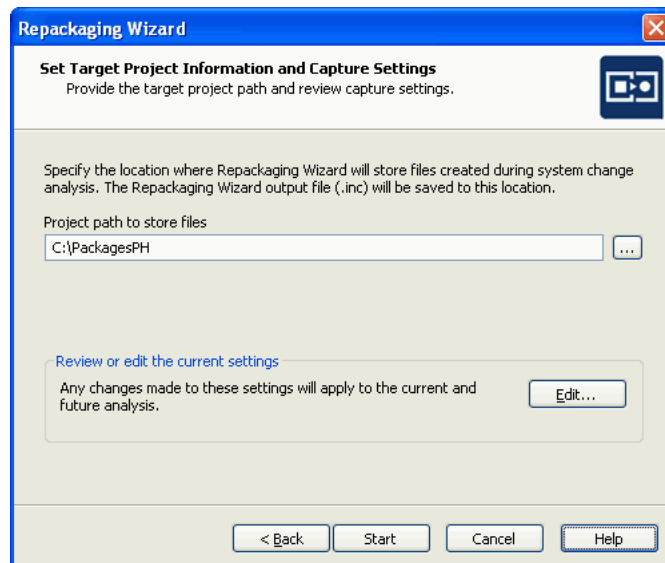
The following information was filled in:

Field	Entry
Program File	C:\WINDOWS\system32\msiexec.exe
Command line Argument(s)	/i "C:\DIRECTORYPATH\PACKAGENAME.msi"



Caution • Do not edit the entries in the **Program File** or **Command line Argument(s)** fields.

8. In the **Product Information** area, make entries in the **Product Name**, **Version**, and **Company Name** fields.
9. Click **Next**. The **Set Target Project Information and Capture Settings Panel** opens.



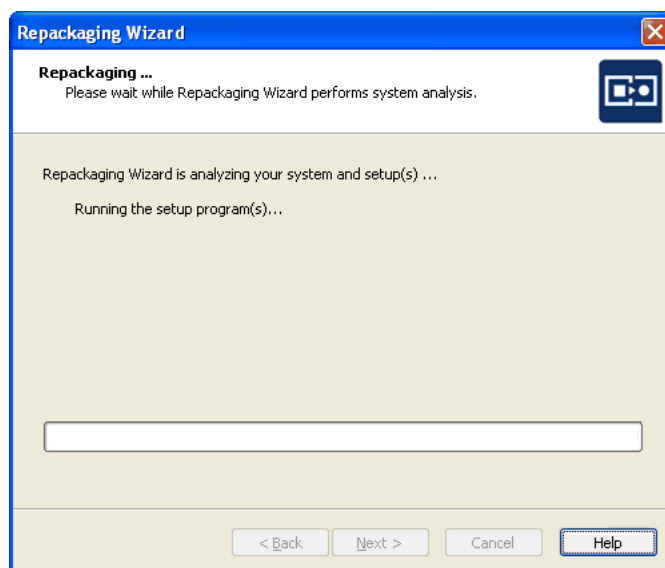
10. Click the Browse (...) button next to the **Project path to store files** field and select the directory where you want the Repackaging Wizard to place its output, including the Repackager project file (.irp), the Repackaging Wizard output files, and source files.

You can also enter the name of a new folder in the **Project path to store files** field, and you will be prompted to create it when you exit this panel.

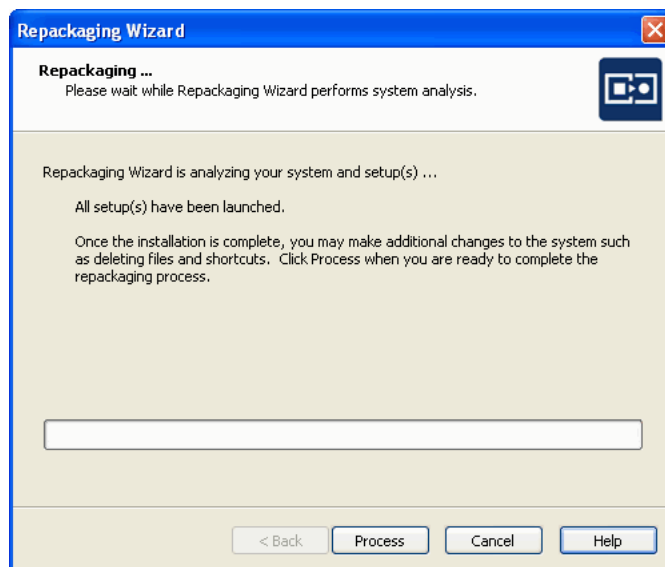
11. To begin the repackaging process, click **Start** on the Set Target Project Information and Capture Settings Panel. The **Repackaging Panel** opens and the Repackaging Wizard captures the initial system status. Then, the selected setup program will be launched.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Legacy Installations Using the Repackaging Wizard

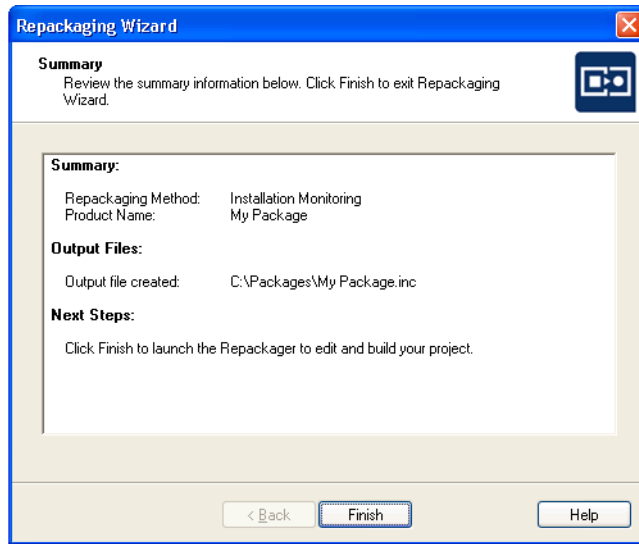


12. Follow the prompts until the installation has completed. When the installation is complete, you are prompted to make any additional changes to the system (such as deleting files and shortcuts) that you want to be recorded in this repackaged installation.



13. When you are ready to complete the repackaging process, click **Process**. The Repackaging Wizard then analyzes the system and setup data that it collected.

Following repackaging, the **Summary Panel** is displayed, providing confirmation that the repackaging was successful.



14. Click **Finish**. Repackager launches and opens the Repackager project file (*.i rp) that you just created.
15. Continue with the instructions in [Working With Repackager Projects](#).

Repackaging Wizard Reference

This section describes each of the dialog boxes and Wizard panels that you might encounter when using the Repackaging Wizard. The help topics in the Repackager Reference are the same detailed documentation that is displayed when you press the F1 key or click the Help button while working in a dialog box.

Reference information is organized as follows:

Table 7-9 • Organization of Repackager Reference Section

Section	Description
Repackaging Wizard	This section provides a panel-by-panel description of the Repackaging Wizard.
Additional Repackaging Wizard Dialog Boxes	This section describes the dialog boxes that can be accessed from the Repackaging Wizard.
Repackaging Wizard Command-Line Options	This section lists the command-line options that are supported by the Repackaging Wizard.
Reboot Handling in the Repackaging Wizard	This section describes how the Snapshot Method and Installation Monitoring Method handle required reboots during repackaging.

Repackaging Wizard

Repackager provides the Repackaging Wizard to convert a legacy setup into a Repackager project. Using this Wizard, you can select the repackaging method (either Snapshot or Installation Monitoring), specify the setup(s) you want to repackage, and run the setup(s). When the Repackaging Wizard has finished its analysis, Repackager automatically creates a Repackager project (.irp) file, which can be modified in Repackager. You can then convert this file to an InstallShield Editor project (.ism) for further editing, or convert it directly to a Windows Installer package (.msi).

The Repackaging Wizard includes the following panels:

- [Welcome Panel](#)
- [Method Selection Panel](#)
- [Snapshot Method Panel](#)
- [Collect Product Information Panel](#)
- [InstallScript MSI Identified Panel](#)
- [InstallShield Professional Setup Panel](#)
- [Set Target Project Information and Capture Settings Panel](#)
- [InstallScript MSI Conversion Output Panel](#)
- [Repackaging Panel](#)
- [Summary Panel](#)
- [Additional Repackaging Wizard Dialog Boxes](#)

Welcome Panel

The Welcome panel appears when you first launch the Repackaging Wizard, providing some introductory information about the use of the Wizard, including that it is for use with traditional (non-Windows Installer-based) installations.



Figure 7-1: Repackaging Wizard Welcome Panel

Method Selection Panel

From the Method Selection panel, select the method(s) you want to use for repackaging.

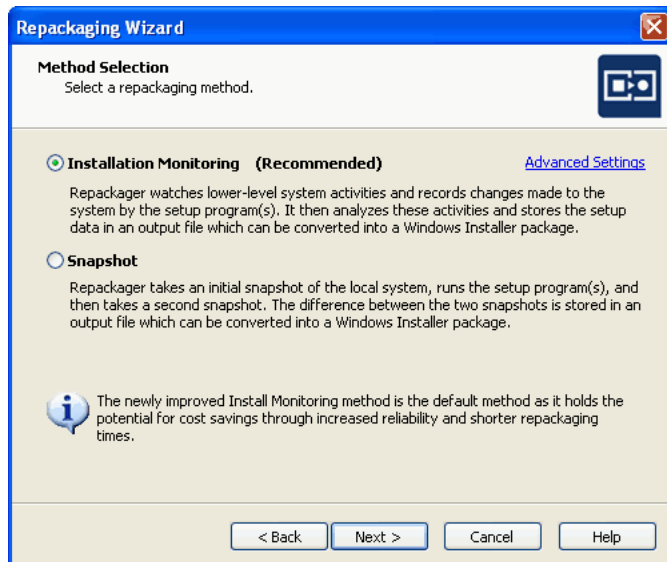



Figure 7-2: Repackaging Wizard Method Selection Panel

The available choices include:

Table 7-10 • Method Selection Panel Options

Options	Description
Snapshot	The Snapshot method involves taking system snapshots before and after an installation, and then creating the Windows Installer package from the difference between them. Any configurations you make between snapshots is also included in the generated Windows Installer package.
Installation Monitoring	<p>Installation Monitoring watches all activities generated by an installation, and then determines the files, .ini files, registry entries and shortcuts that should be included in the generated Windows Installer package.</p> <p>Installation Monitoring is significantly faster than the Snapshot repackaging method.</p> <p>If there are services running on the machine that have nothing to do with the installation being repackaged, click the Advanced Settings link to open the Excluded Processes Dialog Box, where you can choose to exclude those processes.</p> <div><p>Tip • If you know that the installation that you are capturing is from a self-extracting .exe file and if you want to use the Installation Monitoring method, you should click Advanced Settings and add the name of that .exe file to the excluded processes list.</p></div>

Using the InstallShield Professional Logging Method

The InstallShield Professional Logging Method, which should be used to repackage InstallShield Editor and DevStudio 9.x InstallScript installations, is not offered as a choice on the **Method Selection** panel.

However, if you specify an InstallShield Editor or DevStudio 9.x InstallScript installation on the **Collect Product Information** panel (the next panel in the Repackaging Wizard), Repackager will automatically detect the InstallScript installation and will display the **InstallShield Professional Setup** panel. On this panel, you can choose to use the **InstallShield Professional Logging Method** instead of the **Installation Monitoring** or **Snapshot** methods.

It does not matter which option you select on the **Method Selection** panel, but if you select the **Snapshot** method, you must select **Single Step** on the **Snapshot Method** panel (which appears next). If you instead select **Multiple Steps**, Repackager will not recognize the setup as an InstallShield Editor or DevStudio 9.x InstallScript installation and the **InstallShield Professional Setup** panel will not appear.

System Changes Captured by Repackager

Regardless of the repackaging method used, Repackager captures system changes made to the following:

- Application Paths
- Environment Variables

- Files
- INI Files
- NT Services
- ODBC Data Sources
- ODBC Drivers
- Printer Drivers
- Registry Entries
- Shortcuts

If you are using the [InstallShield Professional Logging Method](#) to repackage an InstallShield Editor or DevStudio 9.x InstallScript installation, the following additional information is also collected:

- Path Variables
- Feature/Component Tree

Snapshot Method Panel

The Snapshot Method Panel, which is only displayed if you use the snapshot technology, allows you to specify the way in which you perform repackaging.

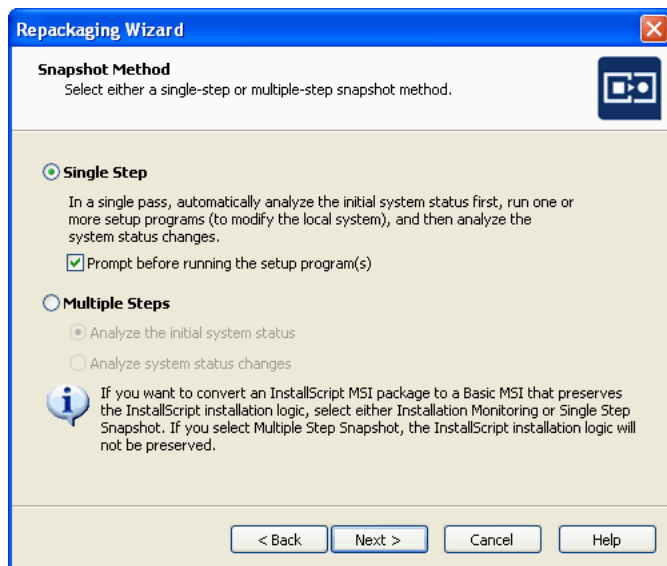



Figure 7-3: Repackaging Wizard Snapshot Method Panel

On the Snapshot Method Panel, you have the following two options:

Table 7-11 • Snapshot Method Panel Options

Option	Description
Single Step	<p>Repackaging in a single step requires you specify at least one setup program to repackage. The Repackager first takes an initial system snapshot, then runs the setup program(s) you specify, and then takes a second snapshot to create the script file that can be converted into a Windows Installer package.</p> <p>You also have the option of requiring the Repackager to prompt you before running the setup program(s), allowing you the opportunity to make changes to your system that you want included in the final package.</p> <div></div> <p>Note • If you are repackaging an InstallShield Editor or DevStudio 9.x InstallScript installation and want to use the InstallShield Professional Logging Method, select Single Step.</p>
Multiple Steps	<p>Repackaging in multiple steps allows you to run the Repackager to obtain an initial system snapshot, after which the Repackager exits. You can then perform any modifications to the system, such as changing configurations, running installations, and so forth. After making the necessary modifications, run the Repackager again to analyze system status changes. The difference between the second Repackager execution and the first results in the script file that ultimately can be converted into a Windows Installer package.</p>

The single step method is very straightforward if you are repackaging applications and not performing many system changes. The multiple step method allows greater flexibility because a setup is not required. This allows you to capture system configurations within the Repackager output, and ultimately within a Windows Installer package. For example, you could modify the screen color depth and create an MSI package for just that configuration.

If Single Step is selected, the **Collect Product Information Panel** is displayed when you click Next. If Multiple Steps is selected and you are performing the initial snapshot, the Collect Product Information panel is displayed, but the Setup Programs area is disabled. If you are performing a system status change analysis, the **Repackaging Panel** appears when you click **Next**.

Collect Product Information Panel

The Collect Product Information panel gathers information necessary for repackaging the installation(s).

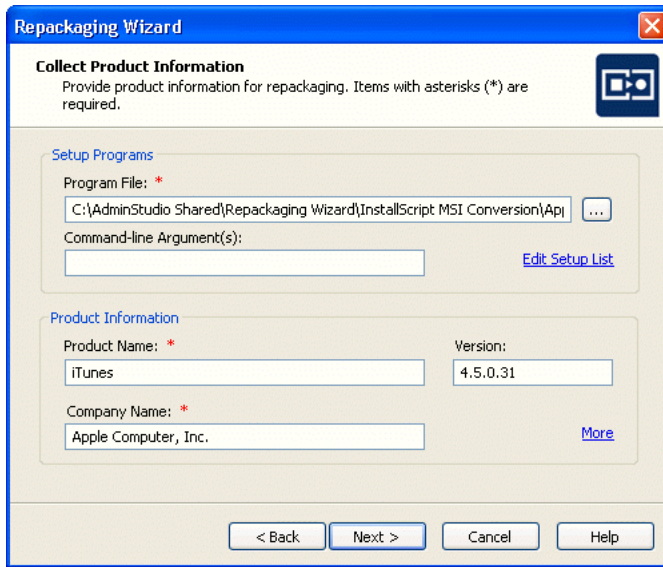


Figure 7-4: Repackaging Wizard Collect Product Information Panel

The information on the Collect Product Information Panel is divided into two sections: **Setup Programs** and **Product Information**.


Setup Programs Area

The Setup Programs area contains information about the setup you are repackaging. Repackager uses this information to launch the setup correctly following pre-analysis. The information collected includes:

Table 7-12 • Setup Programs Options

Properties	Description
Program File	The name and location of the setup executable. Click the Browse (...) button to locate this file. This is a required field.
Command-Line Argument(s)	Any command-line arguments to be used when the setup is run.

Table 7-12 • Setup Programs Options (cont.)

Properties	Description
Edit Setup List	<p>Click to display the Additional Setup Programs dialog box, from which you can enter additional installations to repackage together with this installation. Additional setups share the same product name, version number, and company name in the repackaged installation. However, as you locate each additional setup to repackage, you can specify command-line parameters pertaining only to that setup. You can also specify the order in which the installations are run, should it be necessary.</p> <div><p>Caution • If you are repackaging an InstallShield Editor or DevStudio 9.x InstallScript installation and want to use the InstallShield Professional Logging Method, specify only one Program File. If you specify multiple setup program files, the InstallShield Professional Setup Panel will not appear, and the Repackaging method that you chose on the Method Selection Panel will be used instead.</p></div>

Product Information Area

In the Product Information area, you identify the repackaged installation's **Product Name**, **Version Number**, and **Company Name**.

Table 7-13 • Product Information Options

Field	Description
Product Name	Enter the name for final repackaged installation. This could be the name of the original installation (for example, Tuner), the name of a collective group of products (for example, Microsoft Applications), or another name of your selection (for example, My Apps). This is a required field.
Version Number	Enter the version of the product.
Company Name	Enter the name of the company.

Product Support Information

If you want to associate Web sites with this installation, click the **More** link in the Product Information area to open the Additional Product Information dialog box, where you can enter the **Product URL** and **Support URL** for the application you are repackaging.

InstallScript MSI Identified Panel

This panel opens if the Repackaging Wizard identifies an installation as an InstallScript MSI installation created with InstallShield Editor, InstallShield DevStudio, or InstallShield Developer.

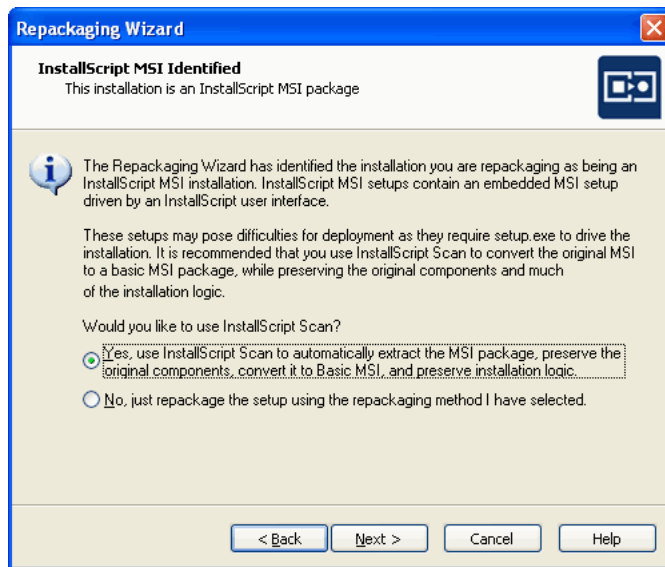


Figure 7-5: Repackaging Wizard InstallScript MSI Identified Panel

InstallScript MSI installations use a Windows Installer database for storage of all file/registry information, but the actual user interface, and much of the installation logic is driven by the InstallScript engine via a `setup.exe` file. This type of installation architecture can cause difficulties during deployment, such as:

- inability to customize or transform the application
- inability to conflict detect
- inability to suppress the user interface
- difficulty patching or upgrading the application

Also, if an InstallScript MSI installation is repackaged using traditional methods (OS Snapshot or Installation Monitoring), significant platform-specific or custom installation, maintenance, and uninstallation logic, and user interface information is lost because those methods only record the installation activities for the specific platform used during repackaging.

Therefore, it is recommended that you use InstallScript Scan to convert an InstallScript MSI installation to a Basic MSI package with InstallScript support. InstallScript Scan preserves the original components and much of the InstallScript installation logic, architecture, and maintainability of the original installation package.

Select one of the following options:

- **Yes**—Use InstallScript Scan to automatically extract the MSI package and convert it to Basic MSI, while preserving the original components and installation logic. This is the default selection.
- **No**—Repackage the installation using the repackaging method selected on the **Method Selection Panel** (Installation Monitoring or Snapshot).

Click **Next** to proceed.

InstallShield Professional Setup Panel

This panel appears in the Repackaging Wizard if you specified an InstallShield Editor or DevStudio 9.x InstallScript installation in the **Program File** field of the Collect Product Information Panel, and if:

- You specified only one program file on the Collect Product Information Panel, *and*
- You selected the **Installation Monitoring** method on the Method Selection Panel, *or*
- You selected the **Snapshot** method on the Method Selection Panel and then selected **Single Step** on the Snapshot Method Panel.

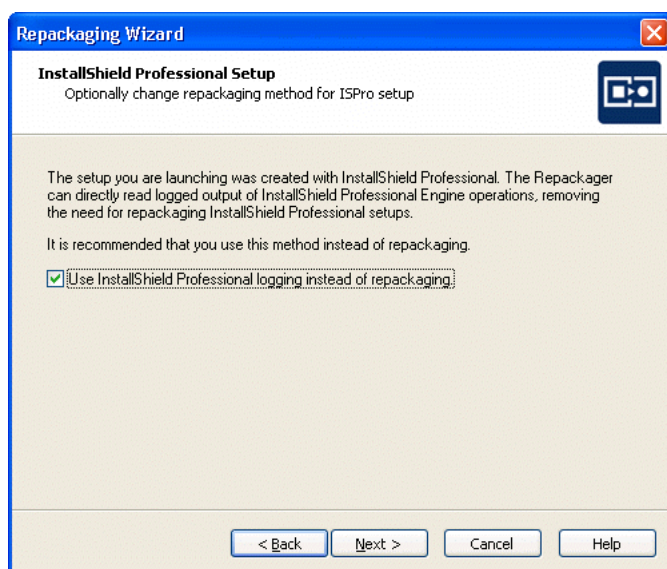


Figure 7-6: Repackaging Wizard InstallShield Professional Setup Panel

If you want to use the InstallShield Professional Logging Method, select the Use InstallShield Professional Logging instead of repackaging check box.

Using the InstallShield Professional Logging Method, Repackager can read logged output of InstallShield Editor and DevStudio 9.x InstallScript installations. This method replaces the other repackaging methods (Installation Monitoring and Snapshot) for InstallShield Editor or DevStudio 9.x InstallScript installations. By using this method, you will be able to get additional information that would not be captured by repackaging, including path variables and a feature tree. For more information, see [InstallShield Professional Logging Method](#).

Set Target Project Information and Capture Settings Panel

The location where you want files created by Repackager stored is defined in the **Project path to store files** field on the **Set Target Project Information Panel**.

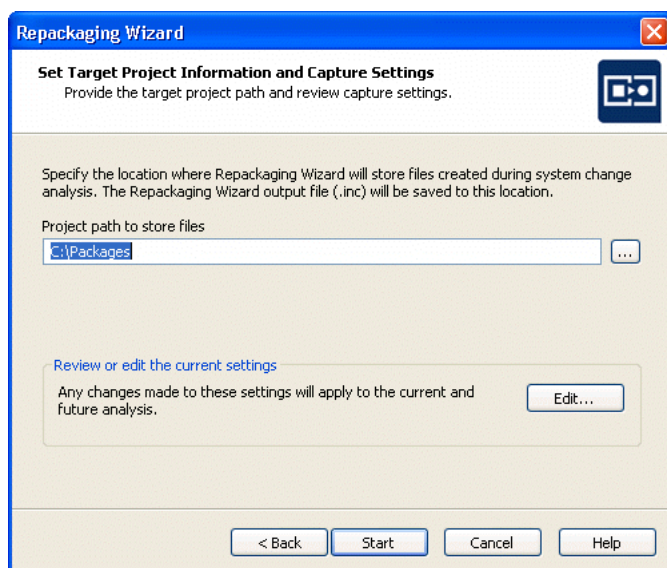


Figure 7-7: Repackaging Wizard Set Target Project Information and Capture Settings

It is recommended that this location not be located on your clean machine, but rather on the same machine as the Repackager executable (most likely on your administrator machine).

You can also review or edit current capture settings by clicking Edit, which displays the **Analysis Options** dialog box.

Click **Start** to begin repackaging and display the **Repackaging Panel**.

InstallScript MSI Conversion Output Panel

On this panel, specify the location where you want Repackager to store the files it creates during InstallScript Scan analysis. The converted Windows Installer MSI package will be saved to this location.

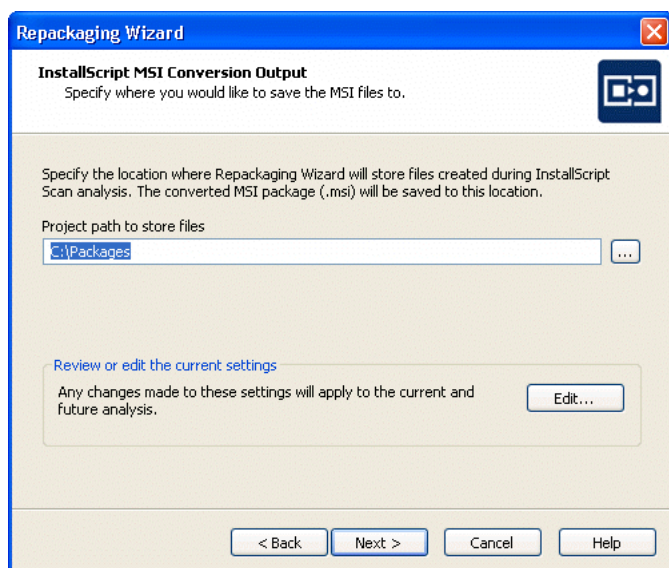


Figure 7-8: Repackaging Wizard InstallScript MSI Conversion Output Panel

It is recommended that this location not be located on your clean machine, but rather on the same machine as the Repackager executable (most likely on your administrator machine).

You can also review or edit current settings by clicking **Edit** to open the **Analysis Options** dialog box. On the **Analysis Options** dialog box, you can specify capture types for the repackaging session, and, for snapshot-mode captures, you can restrict directory analysis to specific directories.

Click **Start** to begin repackaging and display the **Repackaging Panel**.

Repackaging Panel

The Repackaging panel appears while Repackager analyzes your system.

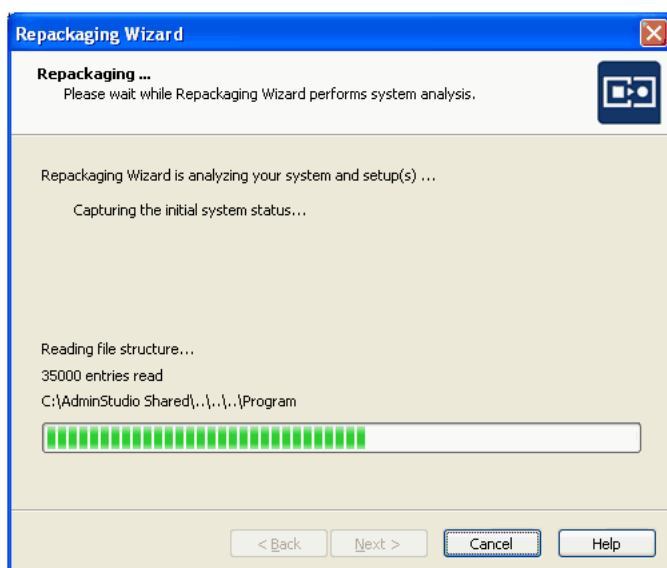


Figure 7-9: Repackaging Wizard Repackaging Panel 1

Depending on settings configured before starting repackaging, the analysis may stop following the initial phase, and again after setup has been run.

After the setups have been completed, you are prompted to click the **Process** button to complete the repackaging process.

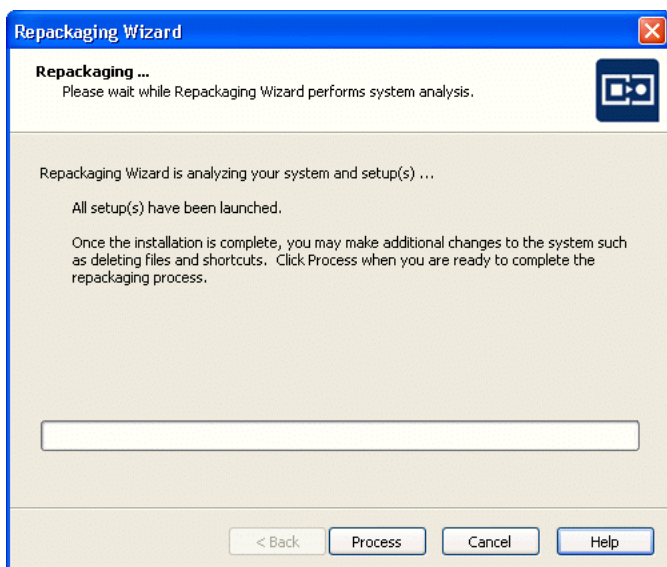


Figure 7-10: Repackaging Wizard Repackaging Panel 2

When you click **Process**, the repackaging is performed and its progress is displayed.

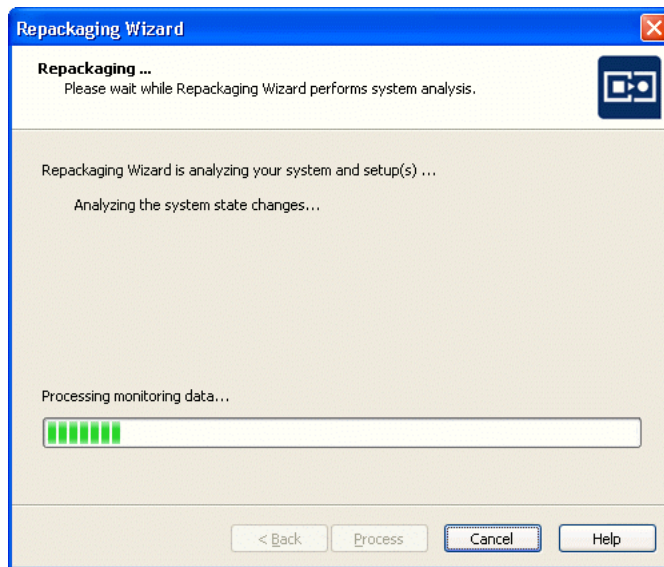


Figure 7-11: Repackaging Wizard Repackaging Panel 3

Following repackaging, the **Summary Panel** is displayed.

Summary Panel

The final panel displayed by Repackager is the Summary panel.

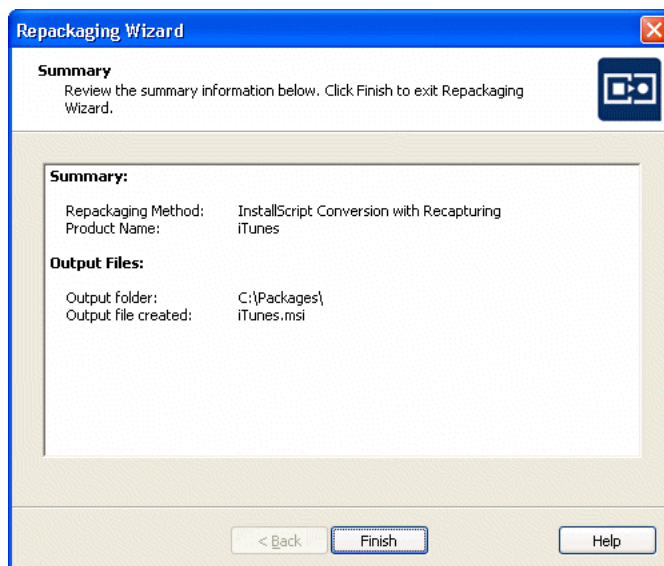


Figure 7-12: Repackaging Wizard Summary Panel

This panel provides confirmation that repackaging was successful, and provides the location of the source setup program(s), the Windows Installer package, and the InstallShield Editor project.

Additional Repackaging Wizard Dialog Boxes

The following dialog boxes can be accessed from the Repackaging Wizard:

- [Additional Setup Programs Dialog Box](#)
- [Setup Information Dialog Box](#)
- [Excluded Processes Dialog Box](#)
- [Analysis Options Dialog Box](#)

Additional Setup Programs Dialog Box

This dialog box, which is accessed by clicking the **Edit Setup List** button on the **Collect Product Information Panel** of the Repackaging Wizard, displays a list of additional setup programs you want to add to the final Windows Installer package.

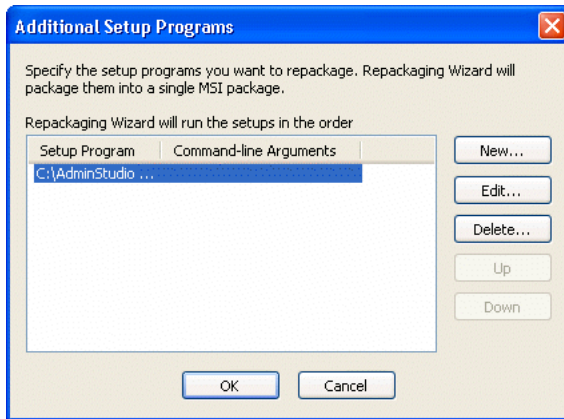


Figure 7-13: Repackaging Wizard's Additional Setup Programs Dialog Box

Essentially, this is a list of the other executables to run, in the order they are to be run, prior to final analysis. The following buttons are available:

Table 7-14 • Additional Setup Programs Dialog Box Buttons

Button	Description
New	Brings up the Setup Information dialog box to enter information about the setup programs.
Edit	Displays the Setup Information dialog box to edit information about the currently selected setup.
Delete	Removes the currently selected setup.
Up	Moves the selected setup up in the setup programs list.

Table 7-14 • Additional Setup Programs Dialog Box Buttons (cont.)

Button	Description
Down	Moves the selected setup down in the setup programs list.

Setup Information Dialog Box

The **Setup Information** dialog box allows you to enter or edit information pertaining to the installations you are repackaging.

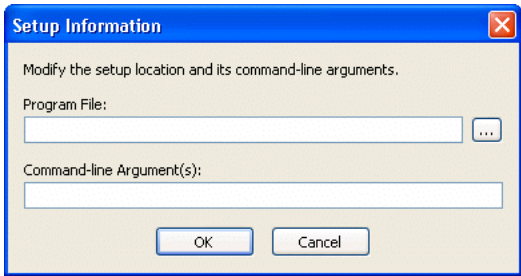


Figure 7-14: Setup Information Dialog Box

Accessible from the **Additional Setup Programs** dialog box, you can provide the name and location of an additional setup program, and any command-line arguments for the setup.

Excluded Processes Dialog Box

During Installation Monitoring, Repackager captures all of the activity of each service or process running on the machine, and then processes this collected data. However, many services running on a machine may have nothing to do with the installation being repackaged. Therefore, you may choose to exclude those processes by adding them to the list on the Excluded Processes dialog box.

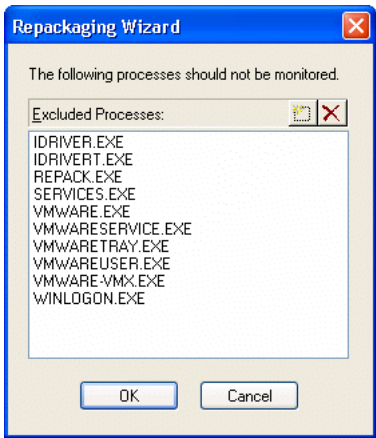




Figure 7-15: Repackaging Wizard Excluded Processes Dialog Box

You can open the Excluded Processes dialog box by clicking the **Advanced Settings** link on the Repackaging Wizard [Method Selection Panel](#). The Excluded Processes dialog box initially lists a default set of processes.

- **To add a process to this list**, click the New () button to add a new blank line to this list, and enter the name of the process that you want to exclude.
- **To delete a process from this list**, select the process and click the Delete () button.

Analysis Options Dialog Box

The Analysis Options dialog box, accessible by clicking Edit from the **Set Target Project Information and Capture Settings Panel** or the **InstallScript MSI Conversion Output Panel**, allows you to specify capture types for the repackaging session.

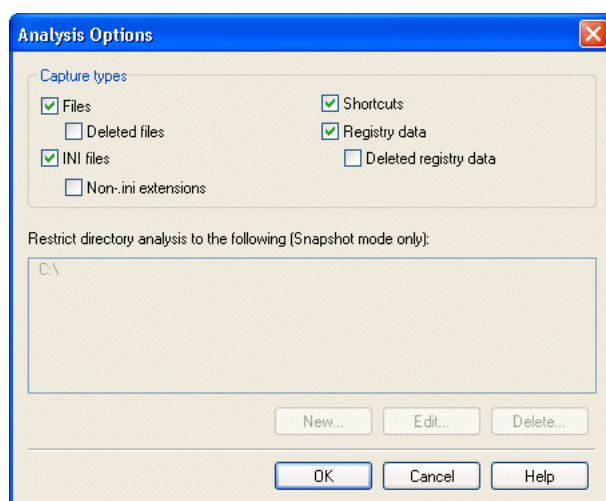


Figure 7-16: Analysis Options Dialog Box

You can select the following:

- Files
- Deleted files
- .ini files
- .ini files with non-.ini extensions
- Shortcuts
- Registry data
- Deleted registry data

Additionally, for snapshot-mode captures, you can restrict directory analysis to specific directories, which can significantly improve repackaging performance. Click **New** to add a directory restriction, **Edit** to modify an existing restriction, or **Delete** to remove a restriction.

Options set in this dialog box apply to the current and subsequent repackaging sessions.

Repackaging Wizard Command-Line Options

The following command-line options are supported by the Repackaging Wizard:

Table 7-15 • Repackaging Wizard Command-Line Options

Option	Description
-?	Displays a dialog box containing usage information for all options.
-app <setup program list>	Allows you to provide a pipe () delimited list of setups to run during repackaging. You can also pass command-line arguments to the setup by separating them from the setup name with a semicolon: -app exe1; cmdline1 exe2; cmdline2 ...exeN; cmdlineN
-b [<project file path and name>.ism]	Specifies the build/release destination folder. If <project file path and name> is not specified, Repackager uses the folder where the .inc file is stored. The project name must end in .ism.
-bv appv	Build App-V application.
-bv thinapp	Build ThinApp application.
-bv citrix	Build Citrix profile.
-cf <config.ini>	This option allows you to select your own configuration template containing exclusions. A sample of this type of file (named Default.ini) can be found in the following directory: <i>AdminStudio Installation Directory\Repackager</i> This particular file contains the default exclusion information.
-cs <configuration type>	This option allows you to select the configuration file type for exclusions. Possible values are: <ul style="list-style-type: none">● Shared—Use shared settings from those stored in the AdminStudio Shared directory.● Custom—Use a custom configuration file (in conjunction with -cf).

Table 7-15 • Repackaging Wizard Command-Line Options (cont.)



Option	Description
-is	<p>Regarding the Repackaging an InstallScript MSI Setup to a Basic MSI Setup procedure, use this parameter in the command line using the following syntax:</p> <pre>Repack.exe -app "c:\setup.exe" -o C:\apps\output -mm -is</pre> <p>In the above example, the user wants to repackage c:\setup.exe using the Installation Monitoring repackaging method (as specified by -mm) and InstallScript conversion (as specified by -is). Repackager would perform the InstallScript conversion process and produce a Basic MSI package with InstallScript support as output. Without the -is parameter, Repackager would perform repackaging without performing InstallScript conversion, and would only create a Repackager .inc file as the output.</p>  <p>Note • The command line parameter -is will be considered only if the setup to be repackaged is a InstallScript MSI setup. If user specified any other legacy setup that is not a InstallScript MSI setup then -is will be ignored.</p>  <p>Note • If user chooses to use the Multiple Step Snapshot repackaging method, then the -is parameter will be ignored. Even if the setup is an InstallScript MSI setup, -is will still be ignored when using the Multiple Step Snapshot repackaging method.</p>
-mode <snapshot mode>	<p>Repackager supports the following repackaging modes for snapshots:</p> <ul style="list-style-type: none"> • single—Single step repackaging that creates an INC file as its output. • pre—Pre-scanning only scans the local drive for a baseline snapshot of the system. • post—Post-scanning only scans the local drive and compares the result with the pre-scan. The differences are written to the INC file as output.
-mm	Instructs Repackager to use installation monitoring as the repackaging technology.
-mp	Instructs Repackager to use the InstallShield Professional Logging Method as the repackaging technology.
-ms	Instructs Repackager to use snapshots as the repackaging technology.
-o <.inc path name>	Specifies a folder path not including the filename. The file name is derived from the Product Name unless overridden with the -of switch.
-of <inc file name>	Specifies the .inc file name that should be used instead of the product name. Use -o to specify the path.

Table 7-15 • Repackaging Wizard Command-Line Options (cont.)

Option	Description
-pc <company name>	Allows you to set the company name.
-pp <product name>	Allows you to set the product name. This will be the same name as the generated Repackager output file (.inc).
-pv <product version>	Allows you to set the product version.
-sb	This option allows you to run Repackager silently, with no user interaction. A progress dialog box is displayed. If no .ini file is specified using the -i parameter, Repackager uses Repack.ini as the default input file. If an output folder is not specified using -o, the default output folder is C:\Packages.
-sn	This option allows you to run Repackager silently, with no user interaction and no progress dialog box. If no .ini file is specified using the -i parameter, Repackager uses Repack.ini as the default input file. If an output folder is not specified using -o, the default output folder is C:\Packages.
-version	This option displays standard version information for Repackager, including the full version and copyright information.



Tip • To open a help topic from the command line that lists command line options, enter the following:

```
repack.exe /?
```



Note • In addition to the - sign for command-line arguments, you can also use the / symbol.

Reboot Handling in the Repackaging Wizard

During repackaging, a setup may require a reboot. For example, some operations may require a file which is in use be replaced, which can only be done after a reboot. Some nuances exist depending on the repackaging technology you are using (Snapshot or Installation Monitoring). In either case, when the Repackaging Wizard detects that a reboot is necessary, the Repackaging Wizard saves the appropriate data and waits until you confirm that you are ready to reboot the machine.

For Snapshot repackaging, the operating system completes the reboot operation. During startup, the operating system restarts all applications and processes and performs any pending file operations. One of the applications that restarts is Repackager. Before you continue processing in Repackager, be patient and ensure all processes and applications have restarted. This may take a minute or two. After the applications and processes have been launched, you can continue repackaging by clicking Process.

For Installation Monitoring, on reboot the operating system launches the Repackaging Wizard, which in turn launches applications and processes and waits until these are finished before prompting you to continue repackaging. However, in some cases the processes or applications launched by the Repackaging Wizard will launch other applications and processes. As in Snapshot repackaging, it is generally a good idea to wait a minute or two before clicking Process.

In both circumstances, waiting helps ensure the setup is fully installed and that captured data contains the necessary information to properly rebuild the setup as an MSI installation.

Chapter 7: Repackaging Legacy Installations Using the Repackaging Wizard

Repackaging Wizard Reference

Converting Legacy Installations Using the Repackager Interface

A Repackager project file (.irp) can be built into an InstallShield Editor project (.ism) or a Windows Installer package (.msi). You can use the Repackager interface to create and modify Repackager project files. You can also use it to build an isolated Windows Installer package and to configure the exclusions used when repackaging a legacy installation.

Information about the Repackager interface is presented in the following sections:

Table 8-1 • Using the Repackager Interface

Section	Description
About the Repackager Interface	Explains how to launch the Repackager interface and how to set options.
Creating Repackager Projects	Explains how to create a Repackager project file (.irp), which can then be built into an InstallShield Editor project (.ism) or a Windows Installer package (.msi).

Table 8-1 • Using the Repackager Interface (cont.)

Section	Description
Working With Repackager Projects	<p>Explains how to build an InstallShield Editor project and Windows Installer package from a Repackager project. The topics in this section include:</p> <ul style="list-style-type: none"> • Building an InstallShield Editor Project • Building a Windows Installer Package • Automatically Generating a Virtual Application During Repackager Project Build • Viewing Repackager Project Properties • Using the Setup Intent Wizard to Detect File Dependencies in a Repackager Project • Creating a Setup Capture Report for a Project • Saving Repackager Projects • Opening InstallShield Editor from Repackager
Isolating Windows Installer Packages	<p>Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested. This section reviews isolation concepts and options, and explains how to build an isolated Windows Installer package.</p>
Configuring Exclusions	<p>Explains how to use Repackager and the Exclusions Editor to configure the exclusions used when repackaging a legacy installation.</p>
Scanning InstallShield Professional Setups for Additional Data	<p>Explains how to scan InstallShield Professional setups for possible additional files.</p>
Creating an InstallShield Editor Template to Use Within Repackager	<p>Explains how to create an InstallShield Editor template to use to speed up the Repackaging process.</p>
Repackager Interface Reference	<p>Describes each of the views and dialog boxes that you might encounter when using the Repackager interface. The help topics in this section are the same detailed documentation that is displayed when you press the F1 key or click the Help button while working in a dialog box.</p>



Note • For information on other Repackager features, see [Repackaging Legacy Installations Using the Repackaging Wizard](#) and [Converting Windows Installer Packages to Virtual Applications](#).

About the Repackager Interface

Information about using the Repackager interface is presented in this section:

- [Launching the Repackager Interface](#)
- [Setting Repackager Options](#)

Launching the Repackager Interface

Repackager can be launched from within the AdminStudio interface. Additionally, if you install Repackager on a network, use Windows Explorer to browse to the `isl1c.exe` executable on the shared drive.



Task: *To launch Repackager from the AdminStudio interface:*

1. Launch AdminStudio.
2. Click the **Tools** tab.
3. From the Tools Gallery, click the **Repackager** icon on the left side.



Repackager

The Repackager Start Page opens and you can begin the repackaging process.



Note • You can also launch Repackager directly from the Windows **Start** menu by pointing to **All Programs**, AdminStudio, **AdminStudio 9.5 Tools**, and clicking **Repackager**.



Caution • It is highly recommended that you repackage applications on a “clean” system. See [Configuring Repackager to Ensure Optimal Installation Capture](#) for more information.

Setting Repackager Options

On the [Options Dialog Box](#), which is opened by selecting **Options** from the **Tools** menu, you can specify the following Repackager options:

- [Selecting Data Display Colors](#)
- [Specifying Additional Merge Module Directories](#)
- [Controlling the Display of ICE Validation Warnings](#)

Selecting Data Display Colors

On the **Colors** tab of the Repackager Options dialog box, you can configure the color of scanned items and deleted items in Repackager's exclusion views (Files, .ini Files, Registry Data, and Shortcuts).



Task: *To change the way excluded and included data is displayed in Repackager:*

1. Open the Repackager interface.
2. From the **Tools** menu, select **Options**. The **Colors** tab of the **Options** dialog box opens.
3. Configure the display colors for **Excluded**, **SmartScan**, and **Setup Intent** items.
4. Click OK.

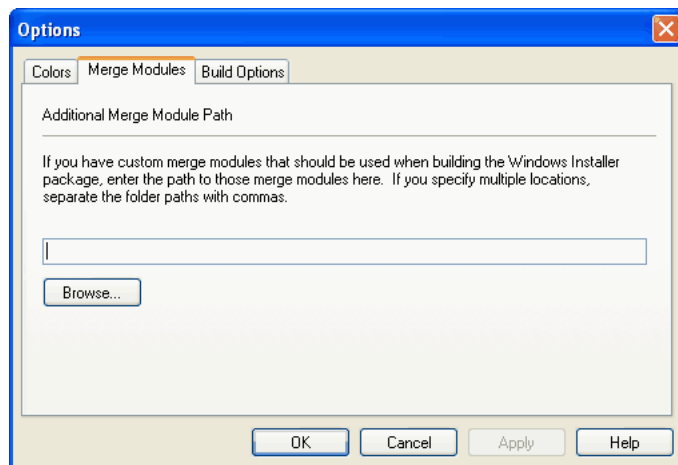
Specifying Additional Merge Module Directories

If you have custom merge modules that should be used when building a Windows Installer package, you need to specify the directories that contain those custom merge modules on the **Merge Modules** tab of the Options dialog box.



Task: *To specify directories of additional Merge Modules:*

1. Open the Repackager interface.
2. From the Tools menu, select **Options**. The **Colors** tab of the Options dialog box opens.
3. Open the **Merge Modules** tab.



4. Enter the directory paths to the custom merge modules. To specify multiple directories, separate the folder paths with commas.



Note • You can click **Browse** and navigate to a directory, but if you browse to a second directory, its directory path will replace the one you initially selected. Therefore, if you want to specify multiple directories separated by commas, you need to manually enter the directory paths.

5. Click **OK**.

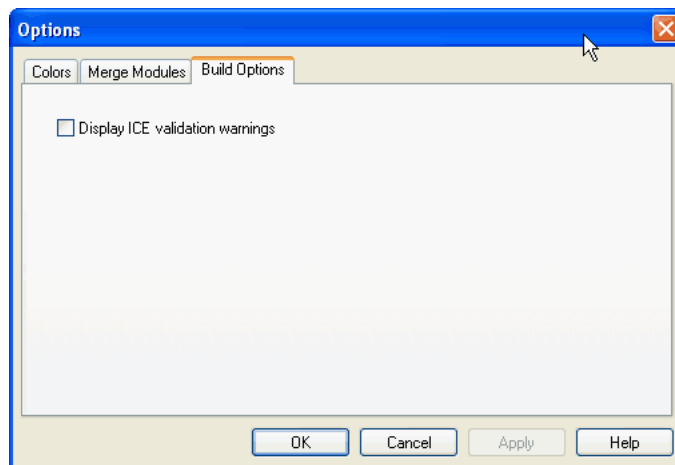
Controlling the Display of ICE Validation Warnings

On the **Build Options** tab of the Options Dialog Box, you can specify whether or not you want to list ICE validation warnings in the Repackager output window during the Build process.



Task: To set the display of ICE validation warnings during builds:

1. From the Repackager interface, select **Options** from the **Tools** menu. The Options dialog box opens.
2. Open the **Build Options** tab.



3. To display any ICE validation warnings that occur during the Repackager Build process, select the **Display ICE validation warnings** option. By default, this option is not selected.

Creating Repackager Projects

Repackager projects (.irp) allow you to visually analyze the files, .ini files, shortcuts, and registry entries captured or changed during the conversion of a legacy setup into a Windows Installer package. You can also exclude files, shortcuts, registry entries, and .ini files from the resulting Windows Installer package, without affecting the original setup data.

There are two methods of creating Repackager projects:

Table 8-2 • Methods of Creating Repackager Projects

Method	Installation Source
Repackaging Wizard	<p>You can use the Repackaging Wizard to convert the following installations:</p> <ul style="list-style-type: none">• InstallShield Professional 1.x to 5.1.x• InstallShield Professional 5.5 to 7.x• InstallShield InstallScript MSI• InstallShield DevStudio 9.x InstallScript• InstallShield Editor InstallScript <p>See Repackaging Legacy Installations Using the Repackaging Wizard.</p>
Repackager Interface	<p>You can use the Repackager interface to convert the following installations:</p> <ul style="list-style-type: none">• Repackager 3.x output (.inc)• Microsoft SMS projects (.ipf)• Novell ZENworks 3.0, 3.1, or 4.0 projects (.axt/.aot)• WinINSTALL projects (.txt) (6.0, 6.5, 7.x)• Wise installation projects (.wse)• InstallShield Professional log files (.isl) <p>See Converting Legacy Installations Using the Repackager Interface.</p>

Converting Legacy Installations Using the Repackager Interface

In addition to repackaging a legacy installation using the Repackaging Wizard, you can also convert many setup types directly to Repackager projects (.irp)—and ultimately to InstallShield Editor projects (.ism) and Windows Installer packages (.msi). Repackager can directly convert the following setup types:

- [Converting Repackager 3.x Output Files](#)
- [Converting a Microsoft SMS Project to a Repackager Project](#)
- [Converting Novell ZENworks Projects](#)
- [Converting WinINSTALL Projects](#)
- [Converting Wise Installation Projects](#)
- [Converting InstallShield Professional Log Files](#)

Converting Repackager 3.x Output Files

To convert a Repackager 3.x output file to a Repackager project, perform the following steps.



Task: *To convert a Repackager 3.x output file (.inc) to a Repackager project (which can subsequently be built into a Windows Installer package):*

1. Launch Repackager.
2. On the **File** menu, click **Open**. The **Open** dialog box opens.
3. Change the **Files of type** filter to **Legacy Repackager Files (*.inc)**.
4. Browse to locate the Repackager 3.x output file you want to convert.
5. Select the file and click **OK**.

The Repackager 3.x project is updated to the Repackager project (.irp) format. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.

Converting a Microsoft SMS Project to a Repackager Project

To convert a Microsoft SMS project to a Repackager project, perform the following steps.



Task: *To convert a Microsoft SMS project (.ipf) to a Repackager project (which can subsequently be built into a Windows Installer package):*

1. Launch Repackager.
2. From the File menu, select Open.
3. In the Open dialog box, change the Files of type filter to SMS Installer (*.ipf).
4. Browse to locate the SMS project you want to convert.
5. Select the project, and click OK.

The legacy project is converted to a Repackager project. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.

Converting Novell ZENworks Projects

You can convert Novell ZENworks 3.0, 3.1, or 4.0 projects (.axt/.aot) to Windows Installer packages (.msi) one at a time or in bulk:

- **Repackager Interface**—You can convert a ZENworks project to a Windows Installer package using the Repackager interface. See [Converting a Novell ZENworks Project Using the Repackager Interface](#).

- **Command Line**—You can use the Command Line to bulk convert multiple ZENworks projects to Windows Installer packages. See [Converting Multiple Novell ZENworks Projects Using the Command Line](#).



Note • In order to convert an .aot file, the ZENworks Desktop Management Agent 6.5 or later (zenlite.dll) must be installed on the workstation where Repackager is installed. If this agent is not installed, Repackager can only convert ZENworks .axt files. See [About .axt and .aot Application Object Template Files](#) for more information.

Converting a Novell ZENworks Project Using the Repackager Interface

Using Repackager, you can convert Novell ZENworks 3.0, 3.1 or 4.0 projects (.axt/.aot) to Windows Installer packages (.msi).

About .axt and .aot Application Object Template Files

In ZENworks Desktop Management, the snAppShot utility generates application object template files—with either an .axt or .aot extension—that contain the details that are required for the Application Launcher to be able to distribute an application to a workstation:

- registry entries to be added
- files to be copied
- changes to be made in the .ini files and system text files (autoexec.bat and config.sys)

Because an .axt file is a text file that can be edited with a text editor in order to modify it after it has been created, it can be opened and converted by Repackager.

However, in order to convert a .aot file (which is not a text file), the ZENworks Desktop Management Agent 6.5 or later (zenlite.dll) must be installed on the workstation where Repackager is installed. If this agent is not installed, Repackager can only convert ZENworks .axt files.



Note • For information on installing the ZENworks Desktop Management agent (version 6.5 or later) to a workstation, see one of the following:

- **Novell ZENworks 6.5 Desktop Management Installation Guide**
<http://www.novell.com/documentation/zenworks65/dminstall/data/front.html>
- **Novell ZENworks 7 Desktop Management Installation Guide**
<http://www.novell.com/documentation/zenworks7/index.html?page=/documentation/zenworks7/dm7install/data/front.html>



Task: **To convert a Novell ZENworks project (.axt/.aot) to a Repackager project (which can subsequently be built into a Windows Installer package):**

1. Launch Repackager.
2. On the **File** menu, click **Open**.

3. In the **Open** dialog, change the **Files of type** filter to either **Novell ZENworks (*.axt)** or **Novell ZENworks (*.axt/*.aot)**.



Note • If the ZENworks Desktop Management Agent 6.5 or later (zenlite.dll) is installed on the workstation where Repackager is installed, the Files of type filter will be Novell ZENworks (*.axt/*.aot). If this agent is not installed, the Files of type filter will be Novell ZENworks (*.axt) and you will be unable to select .aot files as the legacy setup source. See [About .axt and .aot Application Object Template Files](#) for more information.

4. Browse to locate the ZENworks project you want to convert.
5. Select the project, and click **OK**.

The legacy project is converted to a Repackager project. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.

Converting Multiple Novell ZENworks Projects Using the Command Line

To perform a bulk conversion of ZENworks 3.0, 3.1, or 4.0 projects to Windows Installer packages, you use the -Z command line switch.



Task: *To convert multiple Novell ZENworks projects (.axt/.aot) to a Windows Installer package, a Repackager project, or an InstallShield Editor project:*

1. Create an .ini file using the following format:

```
[General]
OutputFormat=MSI|INC|ISM

[AXT]
C:\myData\Project1.axt
C:\myData\Project2.axt
C:\myData\Project3.axt

[AOT]
C:\myData\Project1.aot
C:\myData\Project2.aot
C:\myData\Project3.aot
C:\myData\Project4.aot
```

The following table describes the elements of this file:

Section	Description
[General]	Controls the output format of the entire conversion process. Select one of the following to identify the output format: <ul style="list-style-type: none"> • MSI—Windows Installer package • INC—Repackager output file • ISM—InstallShield Editor project file
[AXT]	List the names and locations of the legacy ZENworks projects (.axt) you want to convert. Include the paths (absolute or relative) to the .axt files.
[AOT]	List the names and locations of the ZENworks .aot projects you want to convert. Include the paths (absolute or relative) to the .aot files.

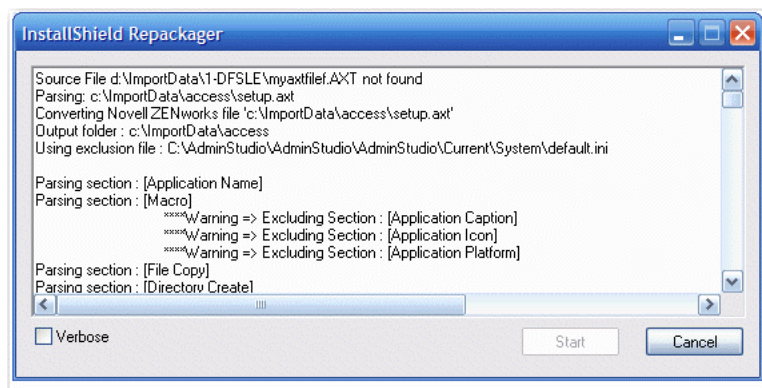
2. Run the repackaging process from the command line using the -Z parameter:

```
ISLC.exe -Z"C:\DirectoryName\FileName.ini"
```



Caution • You must enter a fully qualified path to identify the location of your .ini file.

Repackager loads the .ini file and begins the conversion process. A dialog box opens to display progress messages.



To limit the volume of messages listed, clear the **Verbose** check box.

3. When the repackaging process is complete, the **Cancel** button changes to a **Close** button. Click **Close** to close this dialog box.

You will find the converted files in the location specified in the .ini file as the location of the .aot/.axt input files.

Converting WinINSTALL Projects



Task: *To convert a WinINSTALL 6.0, 6.5, or 7.x project (.txt) to a Repackager project (which can subsequently be built into a Windows Installer package):*

1. Launch Repackager.
2. From the **File** menu, select **Open**.
3. In the **Open** dialog box, change the **Files of type** filter to **WinINSTALL (*.txt)**.
4. Browse to locate the WinINSTALL project you want to convert.
5. Select the project, and click **OK**.
6. If the **WinINSTALL Conversion** dialog box opens, fill in the WinINSTALL-specific variables and click **OK**.

The legacy project is converted to a Repackager project. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.



Note • WinINSTALL projects must be converted to .txt files prior to conversion to Repackager projects.

Converting Wise Installation Projects



Task: *To convert a Wise Installation project (.wse) to a Repackager project (which can subsequently be built into a Windows Installer package):*

1. Launch Repackager.
2. From the File menu, select Open.
3. In the Open dialog box, change the Files of type filter to Wise Projects (*.wse).
4. Browse to locate the Wise Installer project you want to convert.
5. Select the project, and click OK.

The legacy project is converted to a Repackager project. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.

Converting InstallShield Professional Log Files

You can convert an InstallShield Professional log file (.isl) to a Repackager project if you have access to the original setup media. When you open the log file, following the steps below, Repackager will try to find the original setup media automatically (in the location specified in the log file), but if it cannot, it will allow you to browse to it before continuing. If you do not have access to the original setup media, the conversion will fail.



Task: *To convert an InstallShield Professional Log File (.isl) to a Repackager project (which can subsequently be built into a Windows Installer package):*

1. Launch Repackager.
2. From the File menu, select Open.
3. In the Open dialog box, change the Files of type filter to InstallShield Pro Log Files (*.isl).
4. Browse to locate the InstallShield Professional log file you want to convert.
5. Select the file, and click OK.

The log file is converted to a Repackager project. Files, .ini files, shortcuts, and registry entries within the project are visible through the appropriate views in the Repackager Interface.

Working With Repackager Projects

After creating a Repackager project—by [Repackaging Legacy Installations Using the Repackaging Wizard](#) or by [Converting Legacy Installations Using the Repackager Interface](#)—you can perform the following tasks:

- [Building an InstallShield Editor Project](#)
- [Building a Windows Installer Package](#)
- [Automatically Generating a Virtual Application During Repackager Project Build](#)
- [Viewing Repackager Project Properties](#)
- [Using the Setup Intent Wizard to Detect File Dependencies in a Repackager Project](#)
- [Creating a Setup Capture Report for a Project](#)
- [Saving Repackager Projects](#)
- [Opening InstallShield Editor from Repackager](#)

Building an InstallShield Editor Project

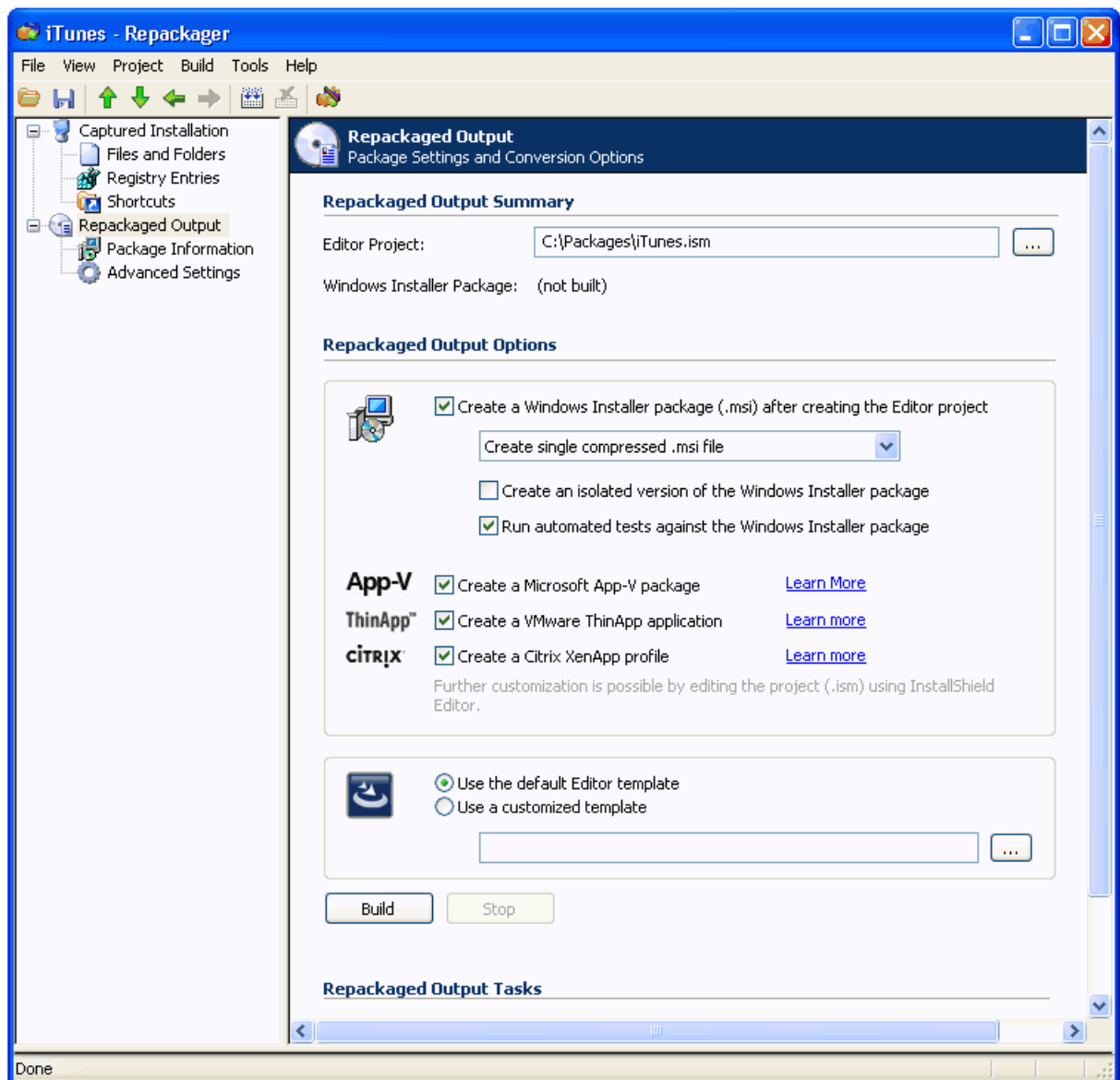
You can build an InstallShield Editor project (.ism) from your Repackager project (.irp).

You can also choose to build just an InstallShield Editor project, so that you can open it in InstallShield Editor and make some modifications prior to building.



Task: *To build an InstallShield Editor project (.ism):*

1. In the Repackager interface, open the Repackager project that you want to convert to an InstallShield Editor project.
2. Select **Repackaged Output** from the View List. The **Repackaged Output View** opens.



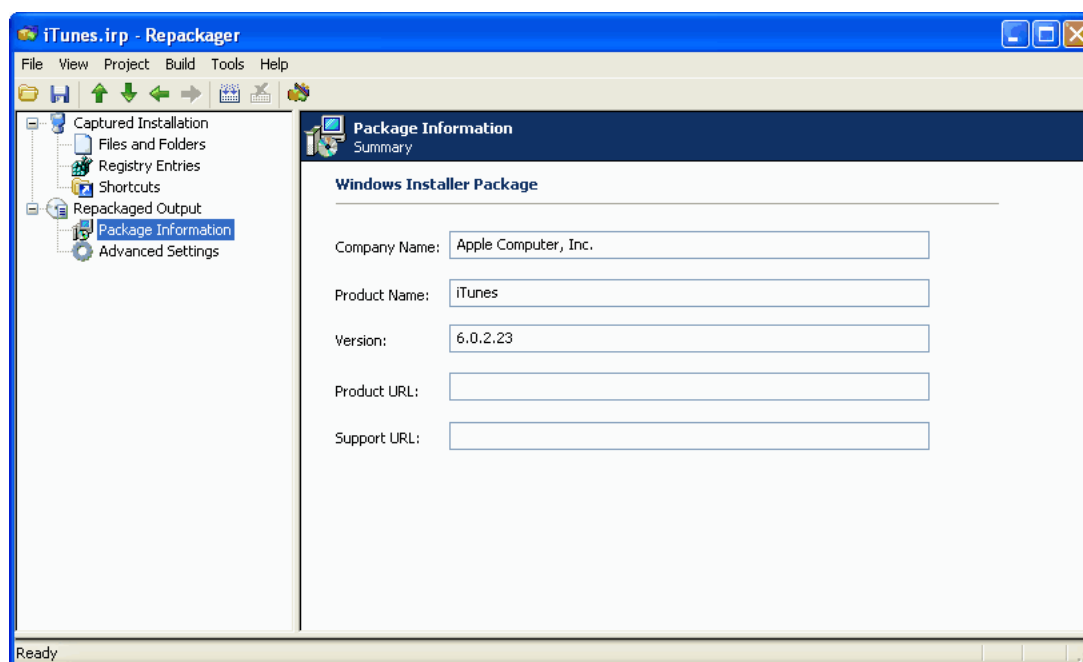
3. In the **Editor Project** field, enter the name and location of the InstallShield Editor Project file you want to create.
4. If you do not want to **Create a Windows Installer package (.msi) after creating the Editor project**, clear this option. If you want to create a Windows Installer Package, see [Building a Windows Installer Package](#).

5. A project template contains all of the default settings and design elements that you want to use as a starting point when you create an installation project. In the **Repackaged Output Options** area, select the InstallShield Editor Project Template you want to use when creating the project:

- **Use the default Editor template**—Select this option to use the default InstallShield Editor Project Template.
- **Use a customized template**—Select this option to use a customized InstallShield Editor Project Template.

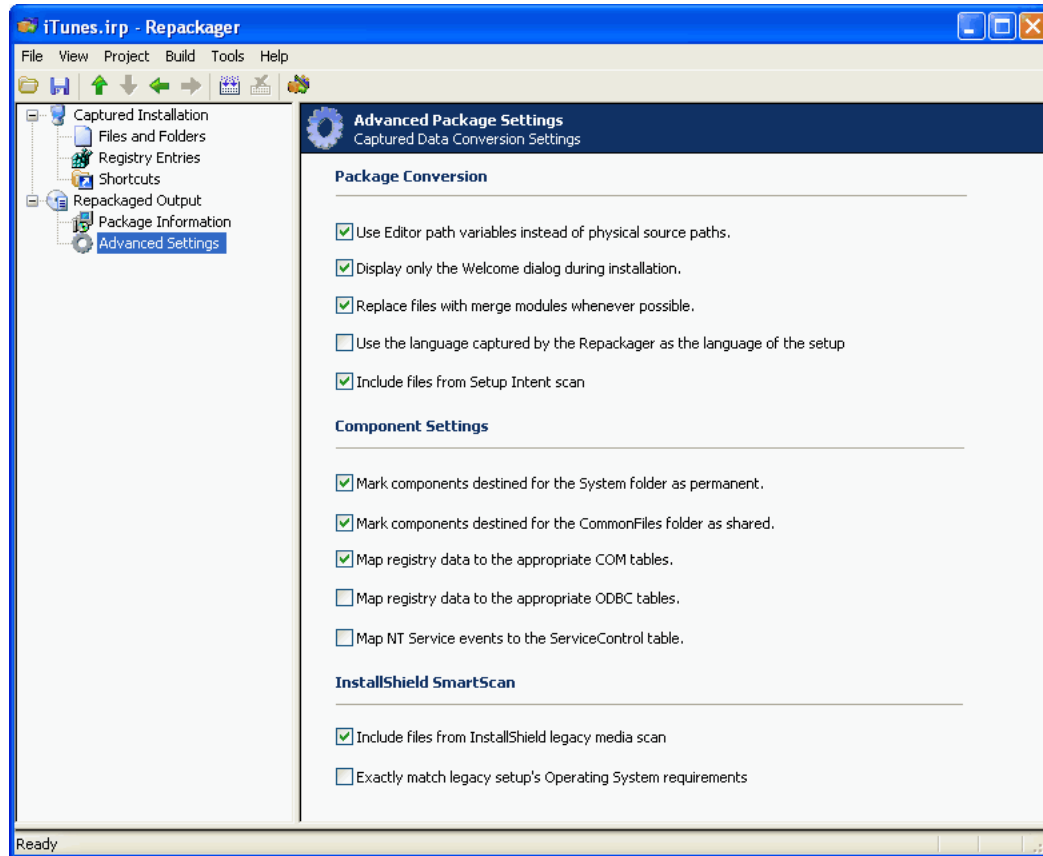
For example, if you wanted all of your InstallShield Editor projects to have a special custom dialog, a set of required redistributables, and a particular SQL script, you could create a project template that has all of those settings. Then, any time that you wanted to create a new project, you could base it off of your custom template. This enables you to avoid re-creating the custom dialog, re-adding the redistributables, and re-adding the SQL script every time that you create a new InstallShield Editor Project.

6. Select **Package Information** from the View List. The Package Information View opens, where you can specify information for the Windows Installer package that you build from the Repackager project. Much of this information may be prepopulated based on settings used in the Repackaging Wizard.



7. Enter the following information:
 - a. **Company Name**—The name of the company that developed the product you are repackaging.
 - b. **Product Name**—The name of the product you are repackaging.
 - c. **Version**—The product's version number.
 - d. **Product URL**—The URL for product information. This appears in **Add/Remove Programs** in the Control Panel.

- e. **Support URL**—A URL for support information. This also appears in **Add/Remove Programs** in the Control Panel, and is often changed during repackaging to provide an internal support URL.
8. Select **Advanced Settings** from the View List. The Advanced Settings View opens.



9. Select the options that you want to use, as described in [Configuring Advanced Conversion Options](#).
10. Select **Repackaged Output** on the View List. The **Repackaged Output View** opens.
11. Click the **Build** button. The build process begins, and its progress is reported in the output window.

When the build process is complete, a [Conversion completed](#) message appears in the output window, and a link to the build log file is provided.

Building a Windows Installer Package

You can simultaneously build an InstallShield Editor project (.ism) and a Windows Installer package (.msi) from your Repackager project (.irp). However, before you do so, you must configure options in your Repackager project necessary for the build.

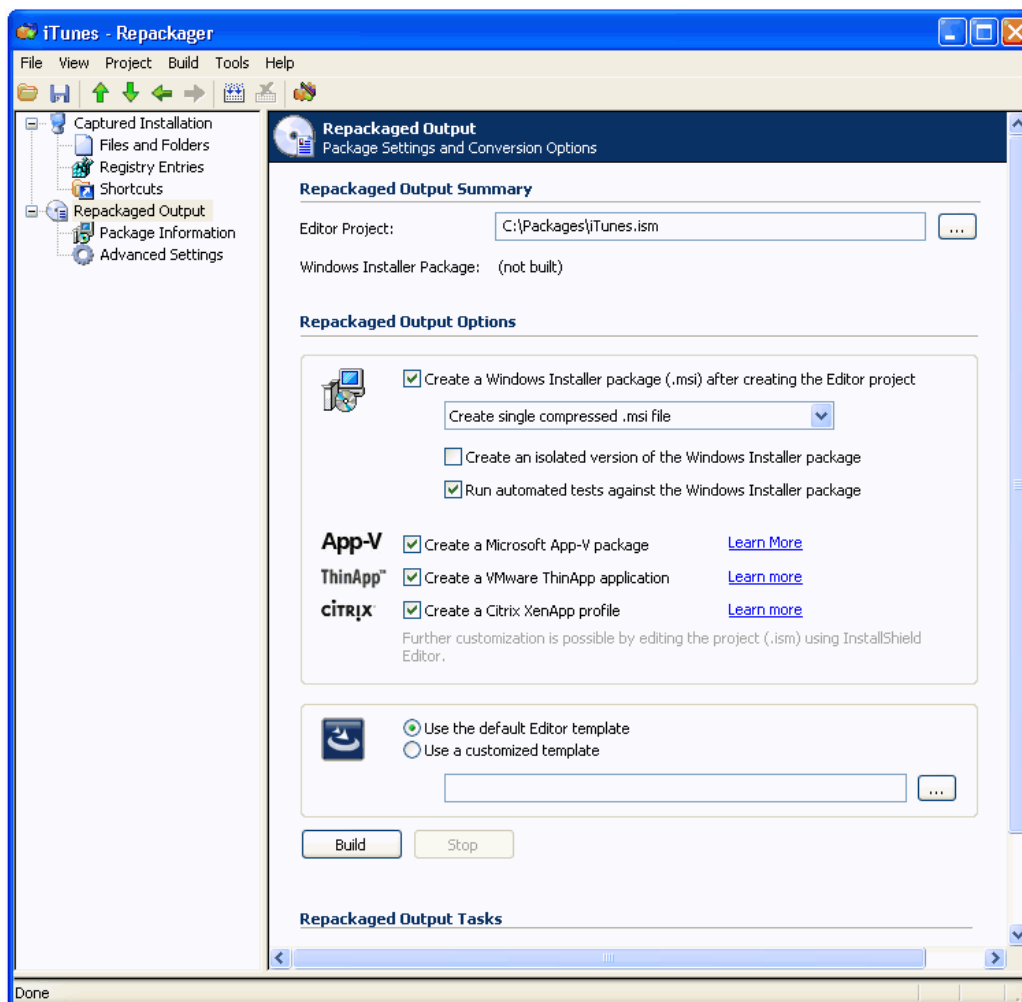


Note • For information on building a virtual application, see [Automatically Generating a Virtual Application During Repackager Project Build](#).



Task: *To build an InstallShield Editor project (.ism) and a Windows Installer package (.msi):*

1. In the Repackager interface, open the Repackager project that you want to convert to an InstallShield Editor project and build a Windows Installer package.
2. Select **Repackaged Output** from the View List. The **Repackaged Output View** opens.



3. In the **Editor Project** field, enter the name and location of the InstallShield Editor Project file you want to create.
4. Select the **Create a Windows Installer package (.msi) after creating the Editor project** option, and select the following additional options:

- a. The compression option that you select for this package depends upon the size of your application's installation and your delivery method.

Neither Setup.exe nor your .msi file can be spanned across multiple disks. So, if the source files associated with your Windows Installer package cannot fit on the same disk as the setup.exe and .msi file, you will need to include them in .cab files on other disks. But if you are performing a network installation and have unlimited space, there is no need to compress files or include additional files in .cab files.

From the list, select one of the following options:

Option	Description
Create single compressed .msi file	Select this option if you want to compress all necessary files inside the .msi package, as opposed to storing them outside of the .msi database.
Create single compressed setup.exe file	Select this option if you want to compress all files inside a setup.exe file, including the .msi file and all other necessary files.
Create .msi file + external compressed .cab file	<p>Select this option if you want to create an .msi file and want to compress the rest of the necessary files in an external .cab file.</p> <p>For example, you might have an installation that contains three features—each containing a 1.5 MB file, Setup.exe, and the installation files for Windows NT—and you want to create a custom media type that is 2 MB in size. The build will span multiple disks.</p> <ul style="list-style-type: none"> • Disk one will contain Setup.exe, InstMsiW.exe (which contains the logic to install the Windows Installer service on Windows NT machines), Setup.ini (which is required for installations that include Setup.exe), and your .msi file. • The remaining disks will contain .cab files that store compressed copies of all your source files.
Create .msi file + external compressed .cab file + setup.exe	Select this option if you want to create an .msi file and a setup.exe file, and want to compress all the rest of the necessary files in an external .cab file.
Create uncompressed .msi file	Select this option if you want to create an uncompressed .msi file. All of the rest of the necessary files, in uncompressed format, would be shipped with the .msi file.
Create uncompressed .msi file and setup.exe	Select this option if you want to create an uncompressed .msi file along with a setup.exe file. All of the rest of the necessary files, in uncompressed format, would be shipped with the .msi and setup.exe files.

- b. To reduce versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested, select the **Create an**

isolated version of the Windows Installer package. An additional Windows Installer package will be created in the same directory as the `.ism` file and the other `.msi` file, with the naming convention of:

`appname.isolated.msi`

For more information on how Repackager isolates applications and the available isolation options, see [Isolating Windows Installer Packages](#).

- c. Select the **Run automated tests against the Windows installer package** option to automatically run PackageExpert tests against the newly built Windows Installer package to determine if it is built according to Windows Installer standards, and if it is in compliance with the installation requirements of the Windows Vista operating system. All of the tests that are currently selected on the PackageExpert **Configuration View** are run, and those selected tests that are automatically resolvable, will be resolved.
- d. To build a virtual application, select the **Create a Microsoft App-V application**, **Create a VMware ThinApp application**, and/or **Create a Citrix XenApp profile** option. See [Automatically Generating a Virtual Application During Repackager Project Build](#).

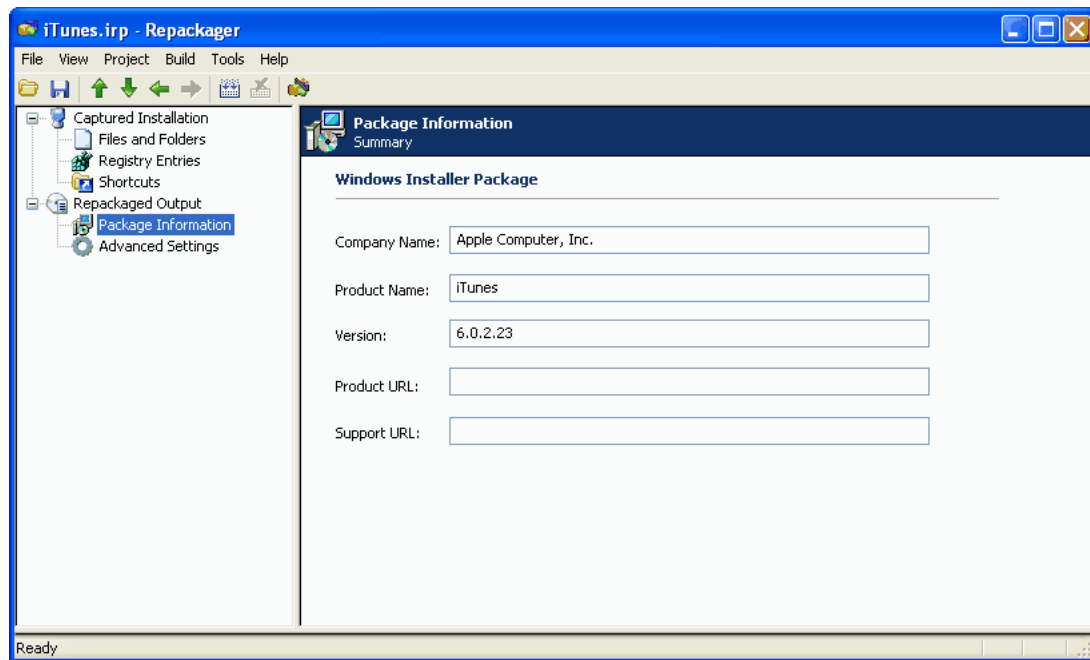


Note • In order to select one of these virtualization options, you must have already selected the **Create a Windows Installer package (.msi) after creating the Editor project** option.

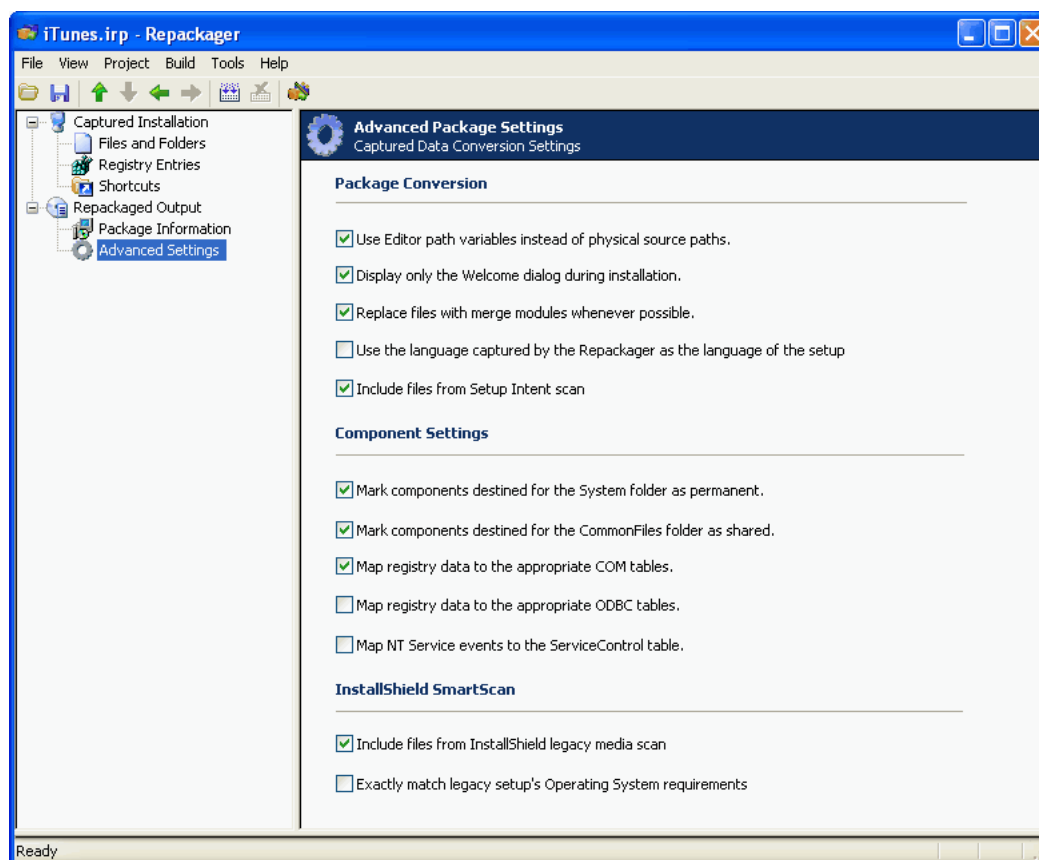
- 5. A project template contains all of the default settings and design elements that you want to use as a starting point when you create an installation project. In the **Repackaged Output Options** area, select the InstallShield Editor Project Template you want to use when creating the project:
 - **Use the default Editor template**—Select this option to use the default InstallShield Editor Project Template.
 - **Use a customized template**—Select this option to use a customized InstallShield Editor Project Template.

For example, if you wanted all of your InstallShield Editor projects to have a special custom dialog, a set of required redistributables, and a particular SQL script, you could create a project template that has all of those settings. Then, any time that you wanted to create a new project, you could base it off of your custom template. This enables you to avoid re-creating the custom dialog, re-adding the redistributables, and re-adding the SQL script every time that you create a new InstallShield Editor Project.

- 6. Select **Package Information** from the **View List**. The **Package Information View** opens, where you can specify information for the Windows Installer package that you build from the Repackager project. Much of this information may be prepopulated based on settings used in the Repackaging Wizard.



7. Enter the following information:
 - a. **Company Name**—The name of the company that developed the product you are repackaging.
 - b. **Product Name**—The name of the product you are repackaging.
 - c. **Version**—The product's version number.
 - d. **Product URL**—The URL for product information. This appears in **Add/Remove Programs** in the Control Panel.
 - e. **Support URL**—A URL for support information. This also appears in **Add/Remove Programs** in the Control Panel, and is often changed during repackaging to provide an internal support URL.
8. Select **Advanced Settings** from the View List. The Advanced Settings View opens.



9. Select the options that you want to use, as described in [Configuring Advanced Conversion Options](#).
10. Select **Repackaged Output** on the View List. The **Repackaged Output View** opens.
11. Click the **Build** button. The build process begins, and its progress is reported in the output window.

When the build process is complete, a [Conversion completed](#) message appears in the output window, a link to the build log file is provided, and the location of the .msi file is listed. For example:

Output file: C:\1516261\WinZip.msi

About the Context.msi File

When some Windows Installer packages are repackaged, some of their data (such as files or registry entries) are excluded according to the normal Repackager exclusion settings. For example, files destined for the \Windows\Installer folder are typically excluded. However, this type of information is occasionally necessary in order to successfully convert a Windows Installer package to a virtual package.

To address this issue, when Repackager builds a Windows Installer package, it now produces two .msi files: packagename.msi and packagename.context.msi.

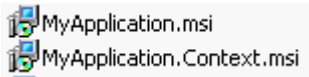


Figure 8-1: Repackaged Output: application.msi and application.context.msi

The .context.msi file contains context data that is necessary in order to convert a .msi file to a virtual package. When creating a virtual package, Repackager combines the data in both the main .msi file and the .context.msi file to produce the final virtual package.

For more information on the .context.msi file, see [Capturing Virtualization Context](#).



Important • If you are not converting a package to a virtual package, you can ignore its .context.msi file.



Note • Context data is not displayed in the Repackager interface when viewing captured Files/Registry details.

Configuring Advanced Conversion Options

To set package conversion, component settings, and InstallShield SmartScan options in your Repackager project, perform the following steps.



Task: *To configure advanced conversion options:*

1. Select **Advanced Settings** from the Repackager View List. The **Advanced Settings View** opens.
2. Under **Package Conversion**, select the package conversion options you want to use during conversion:

Option	Description
Use Editor path variables instead of physical source paths	When storing files in the InstallShield Editor project (.ism), the Wizard uses path variable locations whenever possible.
Display only the Welcome dialog box during installation	Only the Welcome dialog box is displayed when the Windows Installer package is run on a target machine. If this option is unchecked, the default UI sequence is displayed when the setup is installed.
Replace files with merge modules wherever possible	Following best practice rules, Repackager replaces components with comparable merge modules whenever possible.
Use the language captured by the Repackager as the language of the setup	When selected, the target package's language will be the language detected by Repackager (as displayed in the Captured Installation view).

Option	Description
Include files from Setup Intent scan	Any files identified when running the Setup Intent Wizard will be included in the package (unless you have manually excluded them from the project).

3. Under **Component Settings**, select the component settings options you want to use during conversion:

Option	Description
Mark components destined for the System folder as permanent	Executable files installed to the system folder (System32Folder) are marked as Permanent files and will not be uninstalled when the package is uninstalled. This eliminates ICE09 validation errors.
Mark components destined for the CommonFiles folder as shared	Executable files installed to the CommonFilesFolder (or a subfolder of CommonFilesFolder) are marked as shared files. This ensures that these components can coexist with DLLs installed by previous setups.
Map registry data to the appropriate COM tables	Setting this option reduces the number of ICE33 warnings that can occur during package validation, resulting from data not being mapped to the appropriate MSI tables.
Map registry data to the appropriate ODBC tables	If selected, ODBC-related registry data is mapped to ODBC tables instead of the Registry table. This data will only function correctly if Windows Installer supports the ODBC resource being mapped; it is recommended that you do not enable this option if you are unsure whether the ODBC resources are supported correctly by Windows Installer.
Map NT Service events to the ServiceControl table	If selected, NT Service-related registry data is mapped to ServiceControl table instead of the Registry table.

4. Under **InstallShield SmartScan**, select the InstallShield SmartScan options you want to use during conversion:

Option	Description
Include files from InstallShield legacy media scan	Files identified in the Media Scan Wizard will be included in the package (unless you have manually excluded them from the project).

Option	Description
Exactly match legacy setup's Operating System requirements	<p>For SmartScan projects or projects using the Professional Logging Method:</p> <ul style="list-style-type: none">• if this option is selected, component conditions will store specific operating system information: if the filter was NT4, the condition will be (VersionNT=4).• If this option is not selected, component conditions will store a grouping of the operating system: if the filter was NT4, the condition will be (VersionNT). <p>By default, this option is not selected.</p>



Note • For more information on InstallShield SmartScan, see [Scanning InstallShield Professional Setups for Additional Data](#).

Automatically Generating a Virtual Application During Repackager Project Build



Edition • The features on the Repackaged Output View are available in AdminStudio Standard, Professional, and Enterprise Editions.

You can simultaneously build an InstallShield Editor project (.ism), a Windows Installer package (.msi), a Microsoft App-V application, a ThinApp application, and a Citrix profile from your Repackager project (.irp). To do this, you need to select options on the Repackager **Repackaged Output** view.



Task: *To automatically generate a virtual application during Repackager project build:*

1. In the Repackager interface, open a Repackager project.
2. Select **Repackaged Output** from the View List. The **Repackaged Output** view opens.
3. In the **Editor Project** field, enter the name and location of the InstallShield Editor Project file you want to create.
4. Select the **Create a Windows Installer package (.msi) after creating the Editor project** option, and select the associated compression, isolation, and automated test options as described in [Building a Windows Installer Package](#).



Important • When building a virtual package, the **Create a Windows Installer package (.msi) after creating the Editor project** option **must** be selected. If it is not selected, the virtualization options are disabled.

5. Select one or more of the virtual application options:

- **Create a Microsoft App-V application**
- **Create a VMware ThinApp application**
- **Create a Citrix XenApp profile**

The screenshot shows the 'Repackaged Output' dialog box with the title 'Package Settings and Conversion Options'. It is divided into two main sections: 'Repackaged Output Summary' and 'Repackaged Output Options'. In the summary section, the 'Editor Project' is set to '\\schfiler\scratch\PatHowe\Repackager\iTunes.ism' and the 'Windows Installer Package' is '\\schfiler\scratch\PatHowe\Repackager\iTunes.msi'. The 'Repackaged Output Options' section contains several checkboxes. The first checkbox, 'Create a Windows Installer package (.msi) after creating the Editor project', is checked, and a dropdown menu next to it shows 'Create single compressed .msi file'. Below this are three unchecked checkboxes: 'Create an isolated version of the Windows Installer package', 'Run automated tests against the Windows Installer package', 'Create a Microsoft App-V application', 'Create a VMware ThinApp application', and 'Create a Citrix XenApp profile'. Each of the last three has a 'Learn More' link. At the bottom of this section is a note: 'Further customization is possible by editing the project (.ism) using InstallShield Editor.' Below the checkboxes is a section for templates with a radio button selected for 'Use the default Editor template' and an unchecked option for 'Use a customized template' with an empty text field and a browse button. At the bottom of the dialog are 'Build' and 'Stop' buttons. A 'Repackaged Output Tasks' section at the very bottom contains two links: 'Modify the Editor project' and 'Modify the Windows Installer package with Editor'.



Note • If you would like to further customize the virtual application using the InstallShield Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant, you can click the **Modify the Editor project** link below to

open this project in InstallShield Editor. This option is not available until after you build the Repackager project the first time.

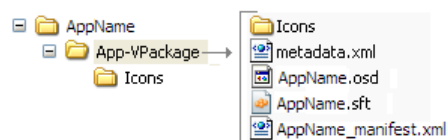


Note • You can also use the Windows Installer Converter to convert a Windows Installer package to a virtual package. See [Converting Windows Installer Packages to Virtual Applications](#).

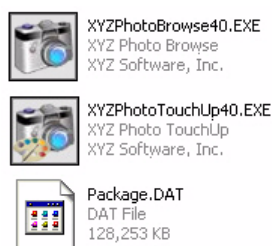
6. Select whether to use the default Editor template or a customized template, as described in [Building a Windows Installer Package](#).
7. Select **Package Information** from the View List and set Package Information options as described in [Building a Windows Installer Package](#).
8. Select **Advanced Settings** from the View List and select the options that you want to use, as described in [Configuring Advanced Conversion Options](#).
9. Select **Repackaged Output** on the View List. The **Repackaged Output View** opens.
10. Click the **Build** button. The build process begins, and its progress is reported in the output window.

When the build process is complete, a [Conversion completed](#) message appears in the output window, and a link to the build log file is provided.

- **If you chose the App-V application option**, a folder named App-VPackage was created in the location you specified in the **Editor Project** field. This folder contains the App-V application for this package and all of its associated files:



- **If you chose the ThinApp application option**, a folder named ThinAppPackage was created in the location you specified in the **Editor Project** field. This folder contains the ThinApp application for this package and all of its associated files:



- **If you chose the Citrix XenApp profile option**, a folder named CitrixProfile was created in the location you specified in the **Editor Project** field. This folder contains the Citrix profile for this package and all of its associated files:



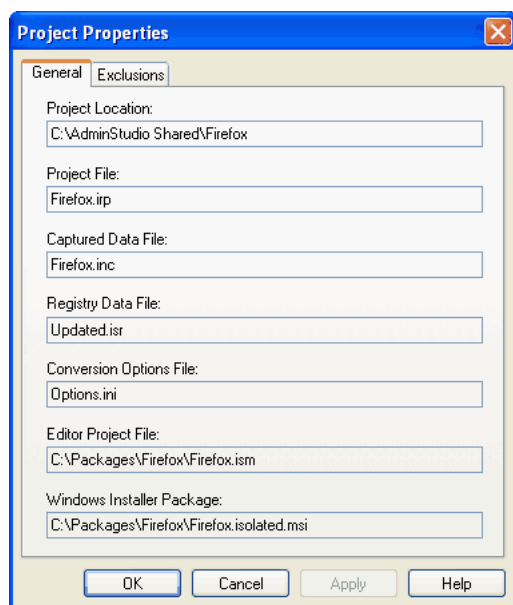
Viewing Repackager Project Properties

You can view the properties of the currently open Repackager project by opening the Project Properties dialog box.



Task: *To view properties for the current Repackager project:*

1. Open a project in the Repackager interface.
2. From the **Project** menu, select **Properties**. The **General** tab of the **Project Properties** dialog box opens.



The following properties are listed:

- **Project Location**—The full path of the directory where the current Repackager project file (.irp) is located.
- **Project File** —The name of the current Repackager project file.
- **Captured Data File**—The name and location of the captured data file (.inc), which was either created by the Repackaging Wizard or during conversion of a Novell ZENworks project, Microsoft SMS project, or WinINSTALL project. The path is relative to the current Repackager project file.

- **Registry Data File**—The name and location of the file containing captured registry data. The path is relative to the current Repackager project file.
 - **Conversion Options File**—The name and location of the `Options.ini` file, which contains an exhaustive list of all options you can use during conversion of the Repackager project to an InstallShield Editor project and Windows Installer package.
 - **Editor Project File**—The name and location of the InstallShield Editor project file as set in the Product View (MSI Package). The path is relative to the current Repackager project file.
 - **Windows Installer Package**—The name and location of the Windows Installer package. The path is relative to the current Repackager project file.
3. When finished viewing properties in the **General** tab, click OK.

Using the Setup Intent Wizard to Detect File Dependencies in a Repackager Project

Although an installation may have intended to install certain files, these files sometimes may not be installed—often because the files already exist on the target machine (either as the same version or a newer version). These files, although not installed or updated, are needed for the product to execute properly when the setup is run on a system that does not already have these files.

You can use the **Setup Intent Wizard** to detect file dependencies that may not be included in your Repackager project (.irp). The **Setup Intent Wizard** scans a setup to identify files that may not have been captured during repackaging—effectively recognizing the installation’s intent for these files.

To use the **Setup Intent Wizard**, perform the following steps:



Task:

To detect file dependencies:

1. From the Project menu, select Setup Intent Wizard. The **Welcome Panel** opens.
2. From the **Welcome** panel, click **Next**. The **Scanning Project Panel** opens.
3. Once scanning is finished, the **Results Panel** opens, listing new files that your setup requires.
4. From the **Results Panel**, select the files you want added to your Repackager project and click **Finish**.
5. Save your Repackager project.



Note • Because the Setup Intent Wizard analyzes files in the Repackager project and searches for dependent files, you must run the Setup Intent Wizard from the same machine where repackaging was performed (with the Repackaging Wizard). You can then save the Repackager project and transfer it to another machine.

Creating a Setup Capture Report for a Project

You can generate an HTML or text document that summarizes the data that was captured when a setup was repackaged.

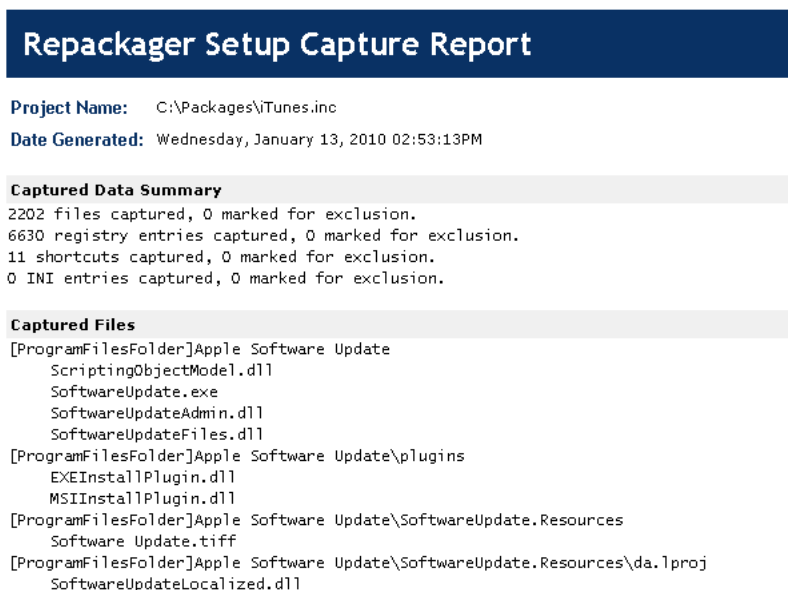


Figure 8-2: Sample Repackager Setup Capture Report

The following information is available to be displayed in this report:

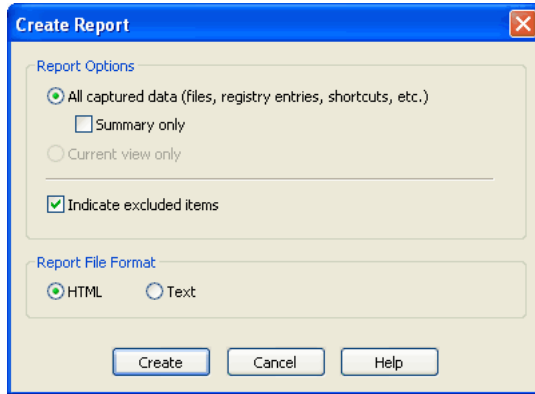
- Captured files
- Captured shortcuts
- Captured .ini file entries
- Captured Registry entries

The report also indicates which of the files, shortcuts, .ini file entries, or Registry entries, if any, have been marked for exclusion. Those marked for exclusion are not included in the Repackager project.



Task: *To create a report detailing captured data:*

1. From the **Project** menu, select **Create Report**. The **Create Report** dialog box opens.



2. Select whether you want the report to contain **All captured data** (all of the data collected during the entire capture), or just the **Current view**.
3. If you want the report to contain data from the entire capture, specify whether you want to just display summary information.
4. Specify whether you want to display excluded items in the report.
5. Select the file format for the report. You can generate an **HTML** report or a **Text** report.
6. Click **Create**. A Save As dialog box opens.
7. From the resulting Save As dialog box, browse to the location where you want to save the file, and provide a name for the report.
8. Click **Save**. The report is saved to the specified location and automatically opens.

Saving Repackager Projects



Edition •

To save a Repackager project, perform the following steps:



Task: *To save the current Repackager project:*

1. Select **Save** from the **File** menu.
or
2. Click the Save button (💾) on the toolbar.



Task: *To save the current Repackager project under a different name:*

Select Save As from the File menu.

Opening InstallShield Editor from Repackager

After building your Repackager project into a Windows Installer package and/or an InstallShield Editor project, you may want to launch InstallShield Editor for additional modifications.



Task: *To launch the generated InstallShield Editor project (.ism) in InstallShield Editor:*

From the Repackager **Project** menu, select **Edit InstallShield Project**. If installed, InstallShield Editor opens the project file.



Task: *To launch the generated Windows Installer package (.msi) in InstallShield Editor:*

From the Repackager **Project** menu, select **Edit Windows Installer Package**. If installed, InstallShield Editor opens the package in Direct MSI Edit mode.

Isolating Windows Installer Packages

Application isolation is one solution to component versioning conflicts. Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.

When building a Windows Installer package from your Repackager project, you can also choose to create an isolated version of that package by selecting an option on the Repackaged Output View.

Information about application isolation is presented in the following topics:

- [About Application Isolation](#)
- [About Assemblies](#)
- [About Manifests](#)
- [About Digital Certificates](#)
- [Setting Isolation Options](#)
- [Building an Isolated Windows Installer Package](#)

About Application Isolation

Application isolation, which is a technique used to minimize the dependencies of an application on system components or dynamic elements, is one solution to component versioning conflicts. Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components and dynamic elements with which it was originally developed and tested.

Isolation is accomplished by:

- Providing DLLs and other shared components for specific applications, *and*
- Placing information traditionally stored in the Registry into other files that specify the locations of these isolated components.

Application isolation provides increased stability and reliability for applications because they are unaffected by changes caused by installation and ongoing maintenance of other applications on the system.

Depending on the isolation options chosen, you can partially or totally isolate an application. When using assemblies and manifests to isolate applications for Windows XP systems, the assemblies can be updated following deployment without necessitating application reinstallation.

Reasons to Isolate Applications

You would want to isolate an application if:

- You want to resolve incompatibilities between different versions of shared components.
- You want to reduce the complexity of the installation by storing COM activation data in a manifest instead of the registry.
- You want to insulate the application from changes to shared components.

Reasons Not to Isolate an Application

You would not want to isolate an application if, following application isolation, you discover that the application no longer works because of an internal dependency on a component that has been moved during the isolation process.



Tip • Following isolation, you can use the *Dynamic Dependency Scanner* in *InstallShield Editor* to verify isolated files are loaded from a different directory.

Isolating Windows Installer Packages Using Application Isolation Wizard

In addition to being able to generate an isolated version of a repackaged setup immediately after the build step in Repackager, you can also use Application Isolation Wizard to isolate a Windows Installer package.

Application Isolation Wizard is a stand-alone tool which accepts a Windows Installer package as input and outputs a new, isolated Windows Installer package.

The Application Isolation Wizard provides a user interface experience that allows the user to extend the initial “dependency scanning” process for identifying file isolation candidates, while in Repackager you specify your assembly and digital signing isolation options on the Isolation Options dialog box, and then those selections are applied to all isolated packages created by Repackager.

For more information, see [Isolating Applications Using Application Isolation Wizard](#).

About Assemblies

Assemblies are DLLs or other portable executable files that applications require to function. Under Windows XP, these can be either shared or private. Private assemblies are typically stored in the same directory as the application they support. Shared assemblies are stored in the WinSxS directory, and are digitally signed.

By creating manifests for assemblies, Repackager allows you to create self-contained applications that can use different versions of the same DLL or other portable executable, without any version conflicts.

Shared Assemblies

Shared assemblies are assemblies available to multiple applications on a computer. Applications that require these assemblies specify their dependence within a manifest. Multiple versions of shared assemblies can be used by different applications running simultaneously.

These assemblies are stored in the WinSxS directory, and must be digitally signed for authenticity. After deployment, the version of shared assemblies can be changed, allowing for changes in dependencies.

Private Assemblies

Private assemblies are assemblies created for exclusive use by an application. They are accompanied by an assembly manifest, which contains information normally stored in the registry. Private assemblies allow you to totally isolate an application, eliminating the possibility that dependent files may be overwritten by other applications.

These assemblies are always stored in the same location as their associated executable.

About Manifests

Manifests, which are used during isolation, are XML files that describe an application. Repackager can create two types of manifests: application manifests and assembly manifest.

Application Manifests Describe an Isolated Application

Application manifests are XML files that describe an isolated application. This descriptive information includes the relationship between the application and its dependent files.

Typically, the naming convention for a manifest is:

`ApplicationName.Extension.manifest`

For example, if the application was `HelloWorld.exe`, the manifest file is called:

HelloWorld.exe.manifest

Assembly Manifests Describe an Application's Assemblies

Assembly manifests are XML files that describe an application's assemblies. This includes components such as DLLs.

Information stored in the assembly manifest, such as COM registration information, ProgIDs, etc., is usually stored in the Registry. However, by making it independent from the Registry, only that application can use the dependent files described in the manifest. This enables you to have multiple versions of the same DLL or other portable executable file on a system without generating compatibility conflicts.

Typically, the naming convention for a manifest is:

AssemblyName.Extension.manifest

For example, if the component was Goodbye.d11, the manifest file is called:

Goodbye.d11.manifest

Manifests as New Components

When you create manifests, Repackager supports putting them into new components. If you do not select the **Create new component for each assembly** option on the **Manifest Options** tab of the **Isolation Options** dialog box, the manifest will be added to the same component as the assembly.

About Digital Certificates

Digital certificates identify you and/or your company to end users to assure them the assembly they are about to use has not been altered. They are issued by a certification authority such as VeriSign, or created using a combination of software publishing credentials (.spc) and a private key (.pvk), both also issued by a certification authority. The certificate includes the public cryptograph key, and, when used in combination with a private key, can be used by end users to verify the authenticity of the signor.

The following digital certificate concepts are defined in this topic:

- [Private Keys](#)
- [Software Publishing Credentials](#)
- [Using a Certificate Store](#)
- [Creating a Certificate File](#)

Private Keys

A private key (a file with the extension .pvk) is granted by a certification authority. Repackager uses the private key you enter in the **Digital Signature** tab of the Isolation Options dialog box to digitally sign your shared assembly and ensure end users of its content's authenticity.

The .spc (Software Publishing Credentials) file and .pvk file you enter in the Digital Signature tab compose the digital certificate for shared assemblies.

Contact a certification authority such as VeriSign for more information on the specifics of software publishing credentials.

Software Publishing Credentials

You must supply a certification authority with specific information about your company and software to obtain software publishing credentials in the form of an .spc file. Your software publishing credentials are used to generate a digital signature for your assembly.

The .spc file and .pvk (private key) file you enter as in the Digital Signature tab of the Advanced Options dialog box compose the digital certificate for shared assemblies.

Contact a certification authority such as VeriSign for more information on the specifics of software publishing credentials.

Using a Certificate Store

To perform code signing, both private key and software publishing credential information must be supplied. This must occur each time a package is signed. Most server operating systems store a certificate locally on the computer that the user used to request the credential information.

Instead of having to store credential files on each of the user computers, you can create a Certificate Store, a storage location which will have numerous certificates, which enables all users or computers with adequate permissions to retrieve the certificate as needed.

Using a Certificate Store allows you to associate the same credentials and private key files with multiple packages. This simplification is particularly useful when isolating applications, as typically the code signing information will be identical for all shared assemblies. Ultimately, the Certificate Store removes the burden of managing private key and software publishing credential information.

Creating a Certificate File

You can create a certificate file from the constituent PVK and SPC files and import it into the Certificate Store using the [PVK Digital Certificate Files Importer](#). You can then export the certificate (.cer) file for use outside of the Certificate Store.



Caution • Certificate files must be 2048-bit or higher. For more information, see the article: [Assembly Signing Example](#) on the Microsoft Developer Network Web site.

Setting Isolation Options

Application isolation is one solution to component versioning conflicts. Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.

On the [Isolation Options Dialog Box](#), which is opened by selecting **Isolation Options** from the **Tools** menu, you can specify the following Repackager isolation options:

- **Assembly Options**—Specify the type of assemblies Repackager will create: private side-by-side assemblies in the application folder or shared side-by-side assemblies in the WinSxS folder. You can also specify the assembly naming conventions. See [Specifying Manifest Options](#).
- **Digital Signature Options**—You can configure the certificate information required when using shared assemblies. This required digital signature provides an extra layer of protection, allowing you to obtain information about the company who created a global assembly. See [Setting Digital Signature Options for Shared Assemblies](#).

Specifying Manifest Options

On the **Manifest Options** tab of the [Isolation Options Dialog Box](#), which is opened by selecting **Isolation Options** from the **Tools** menu, you can specify the following options:

- [Selecting the Assembly Type](#)
- [Specifying the Assembly Naming Conventions](#)



Note • For more information on assemblies and manifests, see [About Assemblies](#) and [About Manifests](#).

Selecting the Assembly Type

On the **Manifest Options** tab of the [Isolation Options Dialog Box](#), which is opened by selecting **Isolation Options** from the **Tools** menu, you can specify the type of assemblies Repackager will create: private side-by-side assemblies in the application folder or shared side-by-side assemblies in the WinSxS folder.



Task: *To select the assembly type:*

1. Open the Repackager interface.
2. From the **Tools** menu, select **Isolation Options**. The **Manifest Options** tab of the **Isolation Options** dialog box opens.
3. Select one of the following **Assembly Type** options:
 - Create private side-by-side assemblies in the application folder.
 - Create shared side-by-side assemblies in WinSxS directory.



Note • Manifests for shared assemblies must be digitally signed. See [Setting Digital Signature Options for Shared Assemblies](#).



Note • The modifications you make on the **Isolation Options** dialog box will be recorded in the *isolationconfig.ini* file, which is stored in the AdminStudio Shared directory.

Specifying the Assembly Naming Conventions

On the **Manifest Options** tab of the [Isolation Options Dialog Box](#), which is opened by selecting **Isolation Options** from the **Tools** menu, you can specify the type of naming conventions Repackager will use when creating assemblies.



Task: *To set the default naming convention for assemblies:*

1. Open the Repackager interface.
2. From the **Tools** menu, select **Isolation Options**. The **Manifest Options** tab of the **Isolation Options** dialog box opens.
3. In the **Assembly Naming Conventions** area, enter your **Company** name and **Division**. These two fields create the default assembly naming convention (in the form Company.Division.Assembly followed by a number).
4. If you want to create a new component for each assembly, select the **Create new component for each assembly** option.

Assemblies created during application isolation will follow the naming convention as specified.



Note • The modifications you make on the **Isolation Options** dialog box will be recorded in the *isolationconfig.ini* file, which is stored in the AdminStudio Shared directory.

Setting Digital Signature Options for Shared Assemblies

You can configure the certificate information required when using shared assemblies on the **Digital Signature** tab of the **Isolation Options** dialog box. This required digital signature provides an extra layer of protection, allowing you to obtain information about the company who created a global assembly.



Note • The modifications you make on the *Isolation Options* dialog box will be recorded in the *isolationconfig.ini* file, which is stored in the AdminStudio Shared directory.




Note • For more information, see [About Digital Certificates](#).



Task: *To set digital signature options:*

1. Open the Repackager interface.
2. From the **Tools** menu, select **Isolation Options**. The **Manifest Options** tab of the **Isolation Options** dialog box opens.

3. Open the **Digital Signatures** tab.
4. Click the Browse () button next to the **Certificate File** field and navigate to the certificate file you are using to sign assemblies.

A digital certificate identifies you and/or your company to end users and assures them the data they are about to receive has not been altered.

5. In the **Code Signing Technology** area, select the type of code signing technology you want to use for the digital signature. You can use one of the following technologies:

Technology	Description
Credentials	<p>Select this option to use credential files as the code signing technology. If you select this option, you must supply the name and location of both your software publishing credential files:</p> <ul style="list-style-type: none">• SPC File—Specify the name and location of your software publishing credentials file (.spc).• PVK—Specify the name and location of your private key file (.pvk). <p>In order to receive a software publishing credentials and a private key, you must supply a certification authority, such as VeriSign with specific information about your company and software.</p>
Certificate Name in the store	<p>Select this option to use the name of an existing certificate file in the Certificate Store as the code signing technology. The Certificate Store is a central repository for certificate files. Using a Certificate Store allows you to reuse the certificate files for different purposes as necessary.</p> <p>As an alternative to providing .spc and .pvk files, you can specify the certificate name as it appears in the certificate store.</p>

Building an Isolated Windows Installer Package

To reduce versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested, select the **Create an isolated version of the Windows Installer package** option on the Repackager **Repackaged Output** view. An additional Windows Installer package will be created in the same directory as the .ism file and the other .msi file, with the naming convention of:

`appname.isolated.msi`

For more information on how Repackager isolates applications and the available isolation options, see [Isolating Windows Installer Packages](#).

Configuring Exclusions

Repackaging exclusions refer to exclusions made during repackage time using the Repackaging Wizard. Any files, registry entries, .ini files, or shortcuts excluded at this point are not included in the Repackager project.

There are two methods of configuring exclusions:

- [Configuring Exclusions Using Repackager](#)
- [Configuring Exclusions Using the Exclusions Editor](#)

Configuring Exclusions Using Repackager

There are three types of exclusions used when repackaging a legacy installation:

Table 8-3 • Repackager Exclusion Types

Exclusion Type	Description
Repackaging Exclusions	<p>Repackaging exclusions refer to exclusions made during repackaging time using the Repackaging Wizard. Any files, registry entries, .ini files, or shortcuts excluded at this point are not included in the Repackager project. Therefore, if you exclude a directory you later need, you need to repackaging the legacy setup again.</p> <p>The Repackager best practice is to capture everything using the Repackaging Wizard, and then exclude visually in the Repackager Interface. This way, you avoid having to run the Repackaging Wizard again if you accidentally exclude necessary files.</p> <p>In some cases, you may want to avoid capturing specific data types during repackaging. For example, your organization may never want to capture shortcuts. You can disable capture of shortcuts during repackaging time, thereby eliminating the need to exclude them later. In Snapshot mode, you may want to limit the analysis to a certain directory to reduce the time it takes to capture the initial and final snapshot.</p>
Project Exclusions	<p>Each Repackager project can use a project exclusion list which marks files, registry entries, shortcuts, and .ini files as excluded in the Repackager project. If your process dictates that you capture everything and only exclude items in the Repackager Interface, then you should set up commonly captured but unnecessary items from the project by default. Because all the data from the original capture is intact, if you accidentally exclude necessary files, you can always reinstate them from the Repackager Interface and quickly rebuild your Windows Installer package.</p>
Individual Project Exclusions	<p>Because each project is different, and may require you to make decisions as to whether certain captured data is necessary, you can also selectively exclude or reinstate items on a per-package basis. These individual project exclusions allow you a fine-level of control as you prepare to build your Windows Installer package from the Repackager project.</p>

Excluding Files

To exclude a captured file from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a captured file from the InstallShield Editor project and Windows Installer package:*

1. Select **Files and Folders** from the View List. The **Files and Folders View** opens.
2. Expand the directory tree and select the directory containing the file you want to exclude.
3. In the file list, right-click the file and then click **Exclude**.

Excluding All Files in a Directory

To exclude all captured files in a directory from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude all captured files in a directory from the InstallShield Editor project and Windows Installer package:*

1. Select **Files and Folders** from the View List. The **Files and Folders View** opens.
2. Expand the directory tree and select the directory containing the files you want to exclude.
3. Right-click the directory and then click **Exclude**.

Excluding Directories and Subdirectories

To exclude all captured files and subdirectories within a directory from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude all captured files and subdirectories within a directory from the InstallShield Editor project and Windows Installer package:*

1. Select **Files and Folders** from the View List. The **Files and Folders View** opens.
2. Expand the directory tree to display the directory containing the files and subdirectories you want to exclude.
3. Right-click the directory and then click **Exclude All**.

Excluding Registry Keys

To exclude a registry key from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a registry key from the InstallShield Editor project and Windows Installer package:*

1. Select **Registry Entries** from the View List. The **Registry Entries View** opens.
2. Expand the Registry tree to display the registry key you want to exclude.
3. Right-click the registry key and then click **Exclude**.

Excluding Registry Values

To exclude a captured registry value from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a captured registry value from the InstallShield Editor project and Windows Installer package:*

1. Select **Registry Entries** from the View List. The **Registry Entries View** opens.
2. Expand the Registry tree and select the registry key containing the value you want to exclude.
3. In the **Registry Value** list, right-click the value and then click **Exclude**.

Excluding .ini Files

To exclude a captured .ini file from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a captured .ini file from the InstallShield Editor project and Windows Installer package:*

1. Select **INI Files** from the View List. The **INI Files View** opens.
2. Expand the **INI Files** tree to display the .ini file you want to exclude.
3. Right-click the .ini file and then click **Exclude**.

Excluding .ini File Sections

To exclude a section in a captured .ini file from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a section in a captured .ini file from the InstallShield Editor project and Windows Installer package:*

1. Select **INI Files** from the View List. The **INI Files View** opens.
2. Expand the **INI Files** tree to display the .ini file containing the section you want to exclude.
3. Right-click the section and then click **Exclude**.

Excluding Shortcuts

To exclude a captured shortcut from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude a captured shortcut from the InstallShield Editor project and Windows Installer package:*

1. Select **Shortcuts** from the View List. The **Shortcuts View** opens.
2. Expand the Shortcuts tree to display the shortcut you want to exclude.
3. Right-click the shortcut and then click **Exclude**.

Excluding All Shortcuts in a Directory

To exclude all captured shortcuts in a directory from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude all captured shortcuts in a directory from the InstallShield Editor project and Windows Installer package:*

1. Select **Shortcuts** from the View List. The **Shortcuts View** opens.
2. Expand the Shortcuts tree to display the directory containing the shortcuts you want to exclude.
3. Right-click the directory and then click **Exclude**.

Excluding Shortcuts from Subdirectories

To exclude all captured shortcuts within a directory or its subdirectories from the InstallShield Editor project and Windows Installer package, perform the following steps.



Task: *To exclude all captured shortcuts within a directory or its subdirectories from the InstallShield Editor project and Windows Installer package:*

1. Select **Shortcuts** from the View List. The **Shortcuts View** opens.
2. Expand the **Shortcuts** tree to display the directory containing the shortcuts and/or subdirectories containing shortcuts you want to exclude.
3. Right-click the directory and then click **Exclude All**.

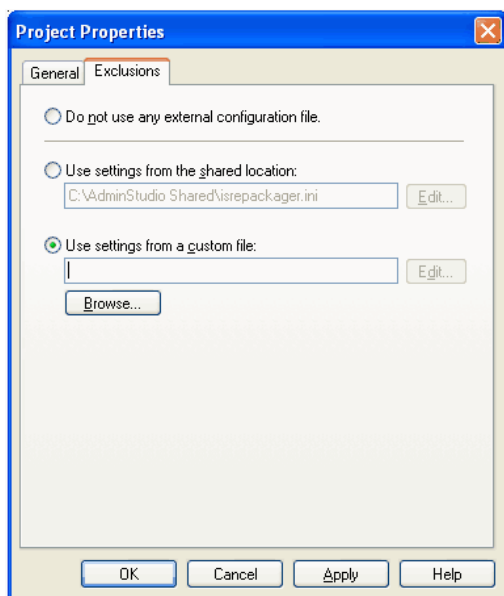
Specifying the External Configuration File

To specify an external configuration file which you want to use as a filter when converting legacy setups, perform the following steps.



Task: *To specify an external configuration file which you want to use as a filter when converting legacy setups:*

1. From the Repackager **Project** menu, select **Properties**. The **General** tab of the **Project Properties** dialog box opens.
2. Open the **Exclusions** tab.



3. Select the **Use settings from the shared location** or the **Use settings from a custom file** option. The **Browse** button for that option is activated.
4. Click **Browse** and select the configuration file you want to use.



Tip • After you select a configuration file, the **Edit** button is activated, enabling you to open the file in the Exclusions Editor.

5. Click **OK**.

When you apply a configuration file, Repackager automatically updates all views to reflect the configuration file's exclusions. However, if you have already excluded items using Repackager, item states are retained.

Modifying External Configuration Files

To configure an external configuration file, perform the following steps.



Task: *To configure an external configuration file:*

1. From Repackager's **Project** menu, click **Properties**. The **General** tab of the **Project Properties** dialog box opens.
2. Select the **Exclusions** tab.
3. Select the file you want to modify in either the **Use settings from the shared location** option or **Use settings from a custom file** options.
4. Click **Edit**. The **Exclusions Editor** opens.
5. Make necessary modifications using the **Exclusions Editor**.
6. When you finish editing the configuration file, click **OK**.
7. Click **OK** to close the **Project Properties** dialog box.

When you apply a configuration file, Repackager automatically updates all views to reflect the configuration file's exclusions. However, if you have already excluded items using Repackager, item states are retained.

Configuring Exclusions Using the Exclusions Editor

The Exclusions Editor allows you to configure three types of exclusions: Repackaging, Project, and OS Snapshot.

Repackaging Exclusions

Repackaging exclusions refer to exclusions made during repackage time using the Repackaging Wizard. Any files, registry entries, .ini files, or shortcuts excluded at this point are not included in the Repackager project. Therefore, if you exclude a directory you later need, you need to repackage the legacy setup again.

The Repackager best practice is to capture everything using the Repackaging Wizard, and then exclude visually in the Repackager Interface. This way, you avoid having to run the Repackaging Wizard again if you accidentally exclude necessary files.

In some cases, you may want to avoid capturing specific data types during repackaging. For example, your organization may never want to capture shortcuts. You can disable capture of shortcuts during repackaging time, thereby eliminating the need to exclude them later. In Snapshot mode, you may want to limit the analysis to a certain directory to reduce the time it takes to capture the initial and final snapshot.

Project Exclusions

Each Repackager project can use a project exclusion list which marks files, registry entries, shortcuts, and .ini files as excluded in the Repackager project. If your process dictates that you capture everything and only exclude items in the Repackager Interface, then you should set up commonly captured but unnecessary items from the project by default. Because all the data from the original capture is intact, if you accidentally exclude necessary files, you can always reinclude them from the Repackager Interface and quickly rebuild your Windows Installer package.

OS Snapshot Exclusions

Like pre-capture repackaging exclusions, you can use the Exclusions Editor to configure exclusions to apply during the capture of OS snapshots. However, to maximize the usefulness of OS snapshots, you should avoid editing the default snapshot exclusion list (ISSnapshot.ini).

Exclusions and Repackager

Exclusions in Repackager refer to files, registry entries, shortcuts, and .ini files that are marked as excluded in the Repackager Interface by default when you open a Repackager project or if you change your exclusions file. The captured data is only marked as excluded and not ignored or discarded during capture. You can create an exclusion file in the Exclusions Editor, and link it to Repackager from the **Exclusions Tab** of the **Project Properties** dialog box in Repackager.

Exclusions and the OS Snapshot Wizard

When using the Exclusions Editor to configure analysis options for capturing OS snapshots, you are creating an exclusion list for files, directories, .ini files, .ini file sections, and registry data. Items in the exclusion list are not captured during the OS snapshot process, and will not be included in the OS snapshot file which is created.

Launching Exclusions Editor

The Exclusions Editor can be launched either within the Repackager interface or outside of Repackager. You can edit the default exclusions file, isrepackager.ini, using either interface.

However, if you want to create a new, custom exclusions file, you must launch the Exclusions Editor outside of Repackager.

- [Launching Exclusions Editor Outside of Repackager](#)
- [Launching Exclusions Editor Within Repackager](#)

Launching Exclusions Editor Outside of Repackager

To launch the Exclusions Editor outside of the Repackager interface, perform the following steps.



Task: *To add a file to the exclusion list:*

1. Launch the Exclusions Editor by locating and executing the following file:
AdminStudio Installation Directory\Repackager\AnalysisOptions.exe
The **Files** tab of the Exclusions Editor opens.
2. Perform one of the following to open an exclusions file:
 - **Shared Exclusions**—To edit the shared exclusions file, on the **Files** menu, point to **Open** and click **Shared Exclusions**. The exclusions in the shared exclusions file are now listed on the **Files** tab.
 - **Custom Exclusions**—To create a new custom exclusions file, on the **Files** menu, click **New**. A default set of exclusions is listed.
3. Make edits to the file.
4. Save the file by selecting **Save** on the **File** menu.
5. If you were creating a custom exclusions file, specify a name and location for this exclusions file and click **Save**.

Launching Exclusions Editor Within Repackager

To add a file to the exclusion list, perform the following steps.



Task: *To add a file to the exclusion list:*

1. Launch Repackager and open a project.
2. On the **Project** menu, click **Properties**. The **Project Properties** dialog box opens.
3. On the **Exclusions** tab, do one of the following:
 - To edit the default exclusions file, select **Use settings from the shared location** and click **Edit**.
 - To edit a custom exclusions file, select **Use settings from a custom file**, browse to the file you want to open (if it is not listed), and click **Edit**.The **Files** tab of the Exclusions Editor opens, with the appropriate exclusions file open.
4. Make edits to the file.
5. Save the file and close the Exclusions Editor by clicking **OK**.



Note • Note that when opening the Exclusions Editor from within Repackager, there is no **File** menu displayed, meaning that you can only edit an existing exclusions file; you cannot create a new exclusions file.

Excluding Files

You use the Exclusions Editor to create an exclusion list for files so that those files are not captured during the OS snapshot process, and will not be included in the OS snapshot file.



Task: *To add a file to the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#).
2. On the **Files** tab, click **New**. The **File Exclusion Information** dialog box opens.
3. Enter or browse to the directory **Path** containing the file you want to exclude.
4. Enter the name of the file you want to exclude, or browse to it by clicking the Browse (...) button to the right of the **Excluded Files** field. If you want to exclude multiple files from the same directory, separate them with pipes (|). If you want to exclude all files in a directory, enter an asterisk (*).
5. Click **OK** to close the **File Exclusion Information** dialog box. The new exclusion appears in the **Files** tab.
6. Save the exclusions file as described in [Launching Exclusions Editor](#).



Note • When configuring file exclusions for Repackager, you are only configuring Repackager to automatically mark the file as excluded; this can be changed from within Repackager on a file-by-file basis. However, when configuring file exclusions for the OS Snapshot Wizard, files in the exclusion list are not captured in the OS snapshot file.

Excluding Files with Specific Extensions

To exclude files with specific extensions from the exclusion list, perform the following steps.



Task: *To exclude files with specific extensions from the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#).
2. On the **Files** tab, click **New**. The **File Exclusion Information** dialog box opens.
3. In the **File Exclusion Information** dialog box, enter or browse to the directory containing the files you want to exclude. If you want to exclude files with a certain extension from all directories, enter an asterisk (*) for the **Path** value.
4. Enter an asterisk followed by the extension you want excluded in the **Excluded Files** field. For example, if you want to exclude all .bak files, enter *.bak. If you want to exclude multiple file types from the same directory (or from all directories), separate each exclusion with a pipe (|).
5. Click **OK** to close the **File Exclusion Information** dialog box. The new exclusion appears in the **Files** tab.

6. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring file exclusions for Repackager, you are only configuring Repackager to automatically mark the file as excluded; this can be changed from within Repackager on a file-by-file basis. However, when configuring file exclusions for the OS Snapshot Wizard, files in the exclusion list are not captured in the OS snapshot file.

Excluding Directories

To add a directory to the exclusion list, perform the following steps.



Task: *To add a directory to the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. On the **Files** tab, click **New**. The **File Exclusion Information** dialog box opens.
3. In the **File Exclusion Information** dialog box, enter or browse to the directory **Path** containing the files you want to exclude.
4. Enter an asterisk (*) in the **Excluded Files** field.
5. Click **OK** to close the **File Exclusion Information** dialog box. The new exclusion appears in the **Files** tab.
6. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring file exclusions for Repackager, you are only configuring Repackager to automatically mark the file as excluded; this can be changed from within Repackager on a file-by-file basis. However, when configuring file exclusions for the OS Snapshot Wizard, files in the exclusion list are not captured in the OS snapshot file.

Editing Existing File Exclusions

To edit an existing file exclusion, perform the following steps.



Task: *To edit an existing file exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Select the appropriate exclusion and click **Edit**. The **File Exclusion Information** dialog box opens.
3. In the **File Exclusion Information** dialog box, modify the **Path** and **Excluded Files** information.

4. Click **OK** to close the **File Exclusion Information** dialog box. The edited exclusion is listed in the **Files** tab.
5. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring file exclusions for Repackager, you are only configuring Repackager to automatically mark the file as excluded; this can be changed from within Repackager on a file-by-file basis. However, when configuring file exclusions for the OS Snapshot Wizard, files in the exclusion list are not captured in the OS snapshot file.

Removing File Exclusions

To remove an existing file exclusion, perform the following steps.



Task: *To remove an existing file exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Select the appropriate exclusion and click **Delete**.
3. Confirm the exclusion by clicking **OK**. The deleted exclusion is removed from the list.
4. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring file exclusions for Repackager, you are only configuring Repackager to automatically mark the file as excluded; this can be changed from within Repackager on a file-by-file basis. However, when configuring file exclusions for the OS Snapshot Wizard, files in the exclusion list are not captured in the OS snapshot file.

Excluding .ini Files

To add an .ini file to the exclusion list, perform the following steps.



Task: *To add an .ini file to the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Open the **INI Files** tab.
3. Click **New**. The **INI File Exclusion Information** dialog box opens.
4. Enter or browse to the .ini file you want to exclude.

5. If there are specific sections you want to exclude from the .ini file, put the section names in brackets ([]) and separate them with pipes (|) in the **Excluded Sections** field. If you want to exclude all sections, put an asterisk (*) in the **Excluded Sections** field.
6. Click **OK** to close the **INI File Exclusion Information** dialog box. The new exclusion appears in the list on the **INI Files** tab.
7. Save the exclusions file as described in [Launching Exclusions Editor](#).



Note • When configuring .ini file exclusions for Repackager, you are only configuring Repackager to automatically mark the .ini file and/or sections as excluded; this can be changed from within Repackager on an .ini file by .ini file basis. However, when configuring .ini file exclusions for the OS Snapshot Wizard, .ini files in the exclusion list are not captured in the OS snapshot file.

Excluding Sections from .ini Files

To add a specific .ini file section to the exclusion list, perform the following steps.



Task: *To add a specific .ini file section to the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#).
2. Open the **INI Files** tab.
3. Click **New**. The **INI File Exclusion Information** dialog box opens.
4. Enter or browse to the .ini file containing the section you want to exclude.
5. To exclude a specific .ini file section, enter the section name in brackets ([]) in the **Excluded Sections** field. If there are multiple sections, separate them with pipes (|).
6. Click **OK** to close the **INI File Exclusion Information** dialog box. The new exclusion appears in the INI Files and Sections Excluded During Analysis dialog box.
7. Click **OK** to close the **INI File Exclusion Information** dialog box. The new exclusion appears in the list on the **INI Files** tab.
8. Save the exclusions file as described in [Launching Exclusions Editor](#).



Note • When configuring .ini file exclusions for Repackager, you are only configuring Repackager to automatically mark the .ini file and/or sections as excluded; this can be changed from within Repackager on an .ini file by .ini file basis. However, when configuring .ini file exclusions for the OS Snapshot Wizard, .ini files in the exclusion list are not captured in the OS snapshot file.

Editing Existing .ini File Exclusions

To edit an existing .ini file exclusion, perform the following steps.



Task: *To edit an existing .ini file exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#).
2. Open the **INI Files** tab.
3. Select the appropriate exclusion and click **Edit**. The **INI File Exclusion Information** dialog box opens.
4. In the **File Exclusion Information** dialog box, modify the **INI File** and **Excluded Sections** information.
5. Click **OK** to close the **INI File Exclusion Information** dialog box. The edited exclusion appears in the list on the **INI Files** tab.
6. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring .ini file exclusions for Repackager, you are only configuring Repackager to automatically mark the .ini file and/or sections as excluded; this can be changed from within Repackager on an .ini file by .ini file basis. However, when configuring .ini file exclusions for the OS Snapshot Wizard, .ini files in the exclusion list are not captured in the OS snapshot file.

Removing .ini File Exclusions

To delete an existing .ini file exclusion, perform the following steps.



Task: *To delete an existing .ini file exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#).
2. Open the **INI Files** tab.
3. Select the appropriate exclusion and click **Delete**.
4. Confirm the exclusion by clicking **OK**. The deleted exclusion is removed from the list.
5. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring .ini file exclusions for Repackager, you are only configuring Repackager to automatically mark the .ini file and/or sections as excluded; this can be changed from within Repackager on an .ini file by .ini file

basis. However, when configuring .ini file exclusions for the OS Snapshot Wizard, .ini files in the exclusion list are not captured in the OS snapshot file.

Excluding Registry Data

To add registry data to the exclusion list, perform the following steps.



Task: *To add registry data to the exclusion list:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Open the **Registry** tab.
3. Click **New**. The **Choose Registry Key** dialog box opens.
4. Enter or browse to the registry key you want to exclude and click **OK**. The key is added to the list on the **Registry** tab.
5. If you want to exclude a certain value in the key, select it from the list and click **Edit**. The **Edit Registry Key** dialog box opens.
6. Provide the **Value Name** you want to exclude, and click **OK** to close the dialog box. The exclusion information is reflected in the list on the **Registry** tab.
7. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring registry exclusions for Repackager, you are only configuring Repackager to automatically mark the registry entry and/or values as excluded; this can be changed from within Repackager on an registry key by registry key basis. However, when configuring registry exclusions for the OS Snapshot Wizard, registry data in the exclusion list is not captured in the OS snapshot file.

Editing Existing Registry Exclusions

To edit existing registry exclusions, perform the following steps.



Task: *To edit an existing registry exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Open the **Registry** tab.
3. Select the registry key that you want to edit and click **Edit**. The **Edit Registry Key** dialog box opens.

4. Modify the exclusion as necessary and click **OK**. The edited information is reflected in the list on the **Registry** tab.
5. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring registry exclusions for Repackager, you are only configuring Repackager to automatically mark the registry entry and/or values as excluded; this can be changed from within Repackager on an registry key by registry key basis. However, when configuring registry exclusions for the OS Snapshot Wizard, registry data in the exclusion list is not captured in the OS snapshot file.

Removing Registry Exclusions

To delete an existing registry exclusion, perform the following steps.



Task: *To delete an existing registry exclusion:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. Open the **Registry** tab.
3. Select the registry key that you want to delete and click **Delete**.
4. Confirm the deletion by clicking **OK**. The deleted exclusion is removed from the list.
5. Save the exclusions file as described in [Launching Exclusions Editor](#):



Note • When configuring registry exclusions for Repackager, you are only configuring Repackager to automatically mark the registry entry and/or values as excluded; this can be changed from within Repackager on an registry key by registry key basis. However, when configuring registry exclusions for the OS Snapshot Wizard, registry data in the exclusion list is not captured in the OS snapshot file.

Repackaging and Anti-Virus Software

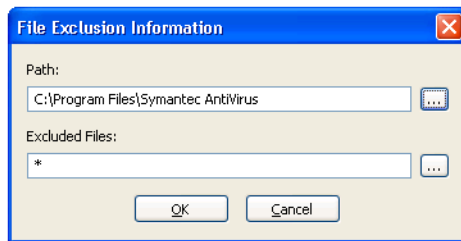
Any machine that you use to repackage most likely has anti-virus software installed on it, even a “clean” machine. During repackaging, the real-time virus detection feature of anti-virus software could automatically update various cached files in its directories.

Therefore, in order to avoid repackaging errors when using the Snapshot repackaging method, you should exclude the software directories containing your anti-virus software.



Task: *To exclude anti-virus software directories:*

1. Launch the Exclusions Editor and open an exclusions file by performing the steps listed in [Launching Exclusions Editor](#):
2. On the **Files** tab, click **New**. The **File Exclusion Information** dialog box opens.



3. Enter or browse to the directory **Path** containing the anti-virus files that you want to exclude. For example, if you wanted to exclude Symantec AntiVirus software, you would select the following directory:

C:\Program Files\Symantec AntiVirus
4. Enter an asterisk (*) in the **Excluded Files** field.
5. Click **OK** to close the **File Exclusion Information** dialog box. The new exclusions appear on the **Files** tab.
6. Save the exclusions file as described in [Launching Exclusions Editor](#):

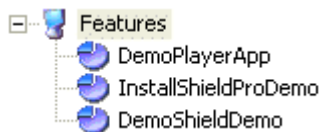


Important • It is strongly recommended that you leave your anti-virus software running during repackaging.

Scanning InstallShield Professional Setups for Additional Data

If you are repackaging a legacy installation that was originally created using InstallShield Professional 5.5 and later, you can use the **SmartScan Wizard** to scan the setup for possible additional files, .ini files, shortcuts, or registry data (such as for language-specific setups or platform-specific setups).

The SmartScan Wizard is also able to group files, shortcuts and registry entries into features corresponding to InstallShield Professional components. Any items (files, folders, shortcuts, or registry entries) that are attached to a component directly or indirectly (through File Groups), are attached to their corresponding features.



Automatic Launch of the SmartScan Wizard

For your convenience, if a Repackaging project is opened (either by using Open on the File menu or by using the Repackaging Wizard), and the project meets certain conditions, the SmartScan Wizard is automatically launched. The SmartScan Wizard is automatically launched:

- if the original setup was an InstallShield Professional 5.5 or later installation, *and*
- the file has not already been repackaged using the **InstallShield Professional Logging Method** (available for InstallShield Editor and DevStudio 9.x InstallScript installations only), *and*
- the SmartScan Wizard has not already been run on this project



Task:

To scan an InstallShield Professional setup:

1. From the **Project** menu, select **SmartScan Wizard**. The **Welcome Panel** appears.
2. Click **Next**. The **Original InstallShield Professional Setup Panel** appears.
3. The **Specify the path of the original setup executable** field will be filled in by default if that information exists. Browse to the InstallShield Professional setup you want to scan, and specify a password if required.
4. In the **Original target folder** field, enter the directory where the product was installed during the repackaging process. In most cases, this will be a subdirectory of [ProgramFilesFolder].
5. Click **Next**. The **Scanning Panel** appears.
6. When scanning is complete, click **Next**. The **Setup Feature Tree Panel** appears, listing the features and components that were configured in the InstallShield Professional project.
7. Select the **Add feature tree** option if you want to use the feature information defined in the Professional media.
8. Click **Next**. The **Scanning Media Panel** appears, showing the results of Repackager's search for any extra setup information in the media.
9. Click **Finish** to apply the results to the project.
 - The setup is scanned, and any additional files, shortcuts, .ini files, or registry data is added to the appropriate view.
 - The data captured by the SmartScan Wizard appears in the color designated in the **SmartScan items** field on the **Colors** tab of the **Options** dialog box.
 - If you selected the **Add feature tree** option, the files, shortcuts and registry entries are grouped into features corresponding to the InstallShield Professional components.



Note • After you have scanned an InstallShield Professional project and included data from it in your Repackager project, you can save the project. When you reopen it, this collected data no longer retains the “scanned” color, since it is now part of the Repackager project.

Creating an InstallShield Editor Template to Use Within Repackager

One of the main reasons you use AdminStudio is to significantly reduce the time it takes to package an application for deployment. You can use the following procedure to speed up the packaging process even more.

You can create an InstallShield Editor template that you can use within the Repackager interface to save additional time when customizing a package. By using this template, all future InstallShield Editor .ism project files generated by Repackager will contain the company-specific default settings that were specified in the template. Using a template is also beneficial for organizations with multiple packagers, since it helps enforce consistency by enabling all packagers to make the same standard customizations to packages.

To create an InstallShield Editor template to use within Repackager, perform the following steps.

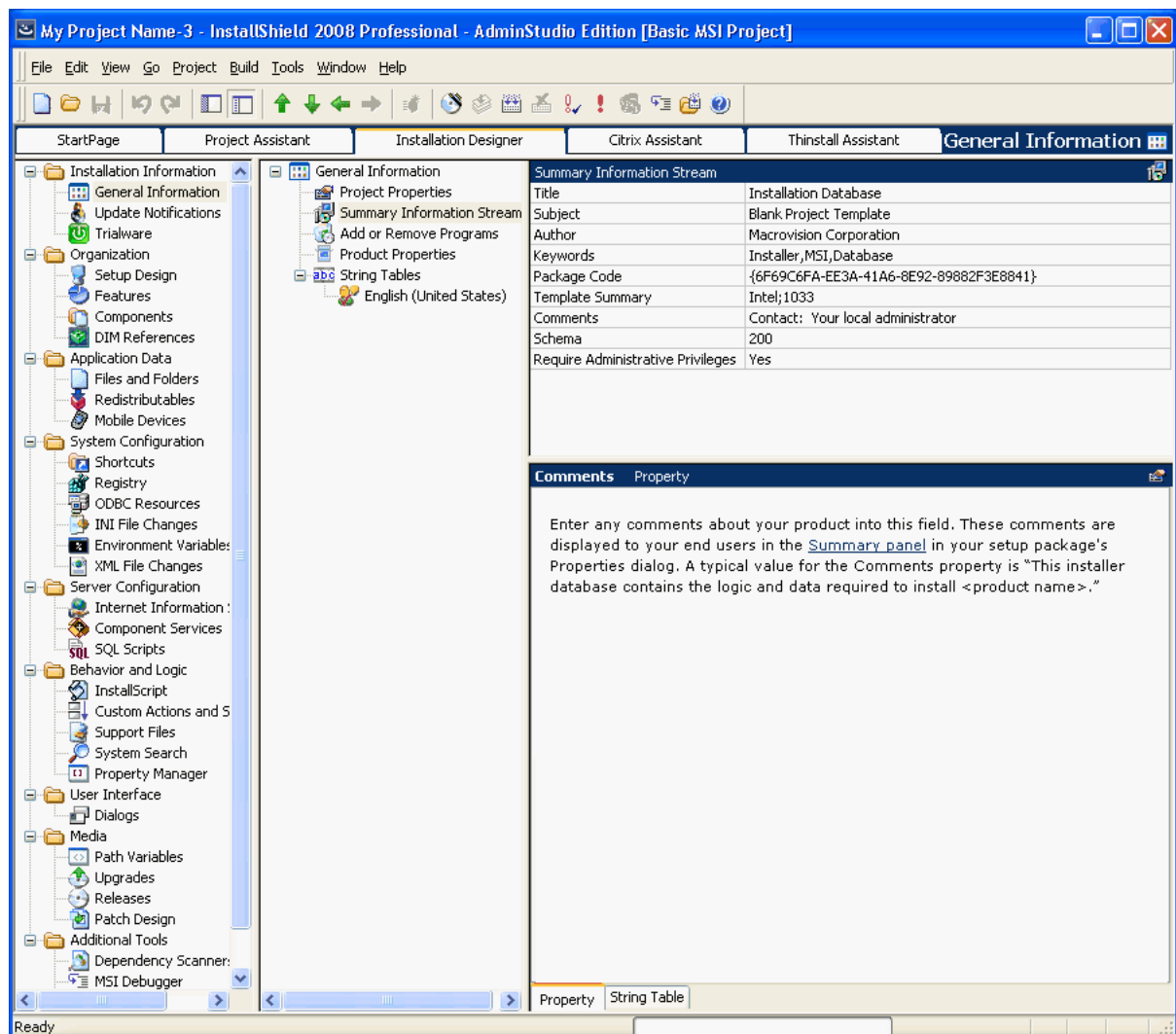


Task: *To create a customized InstallShield Editor template:*

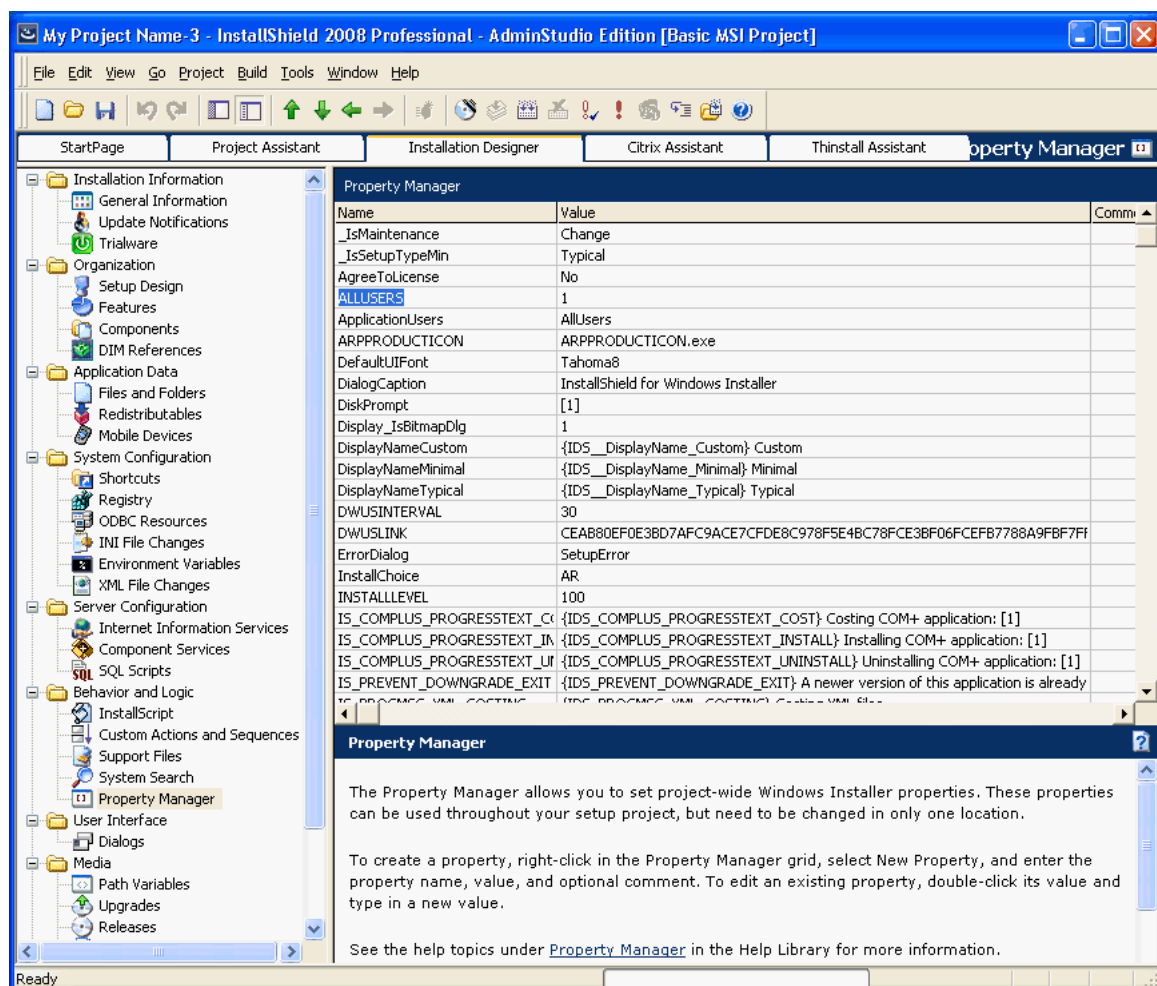
1. Create a new **Basic MSI Project** in the InstallShield Editor.
2. On the **Installation Designer** tab, select the **General Information** node under **Installation Information**, and enter your company-specific information as required.

Chapter 8: Converting Legacy Installations Using the Repackager Interface

Creating an InstallShield Editor Template to Use Within Repackager



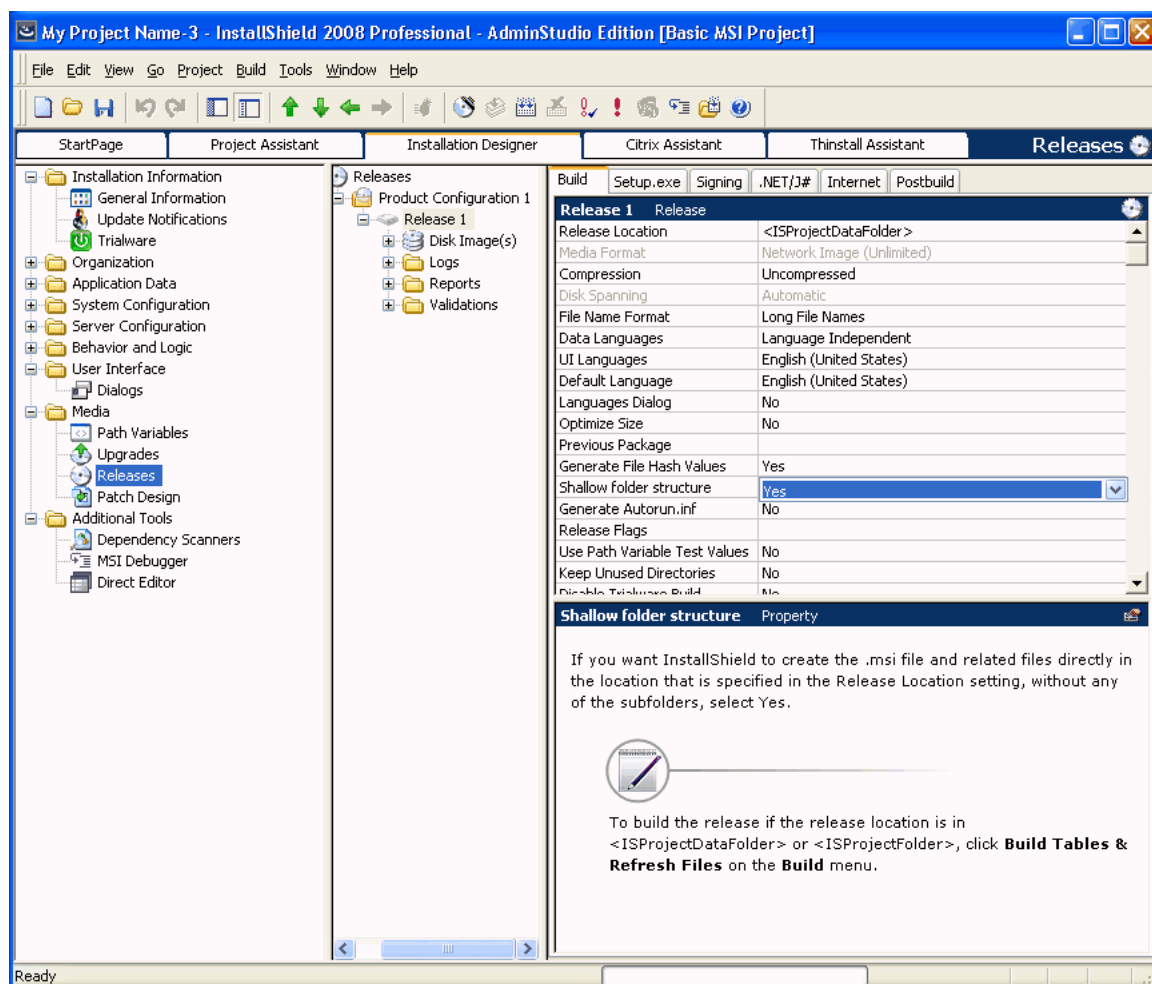
- Under **Behavior and Logic**, select the **Property Manager** node and add the required properties like ALLUSERS, ISSCRIPTDRIVEN, etc.



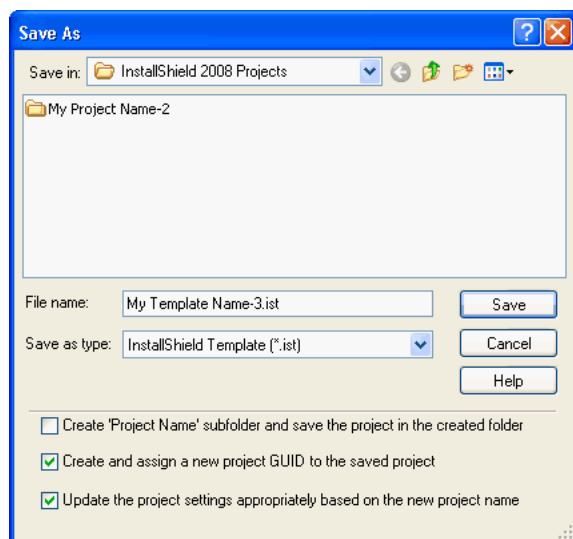
4. You can also optionally set **Shallow Folder Structure** to **Yes** in the **Releases** view under **Media**.

Chapter 8: Converting Legacy Installations Using the Repackager Interface

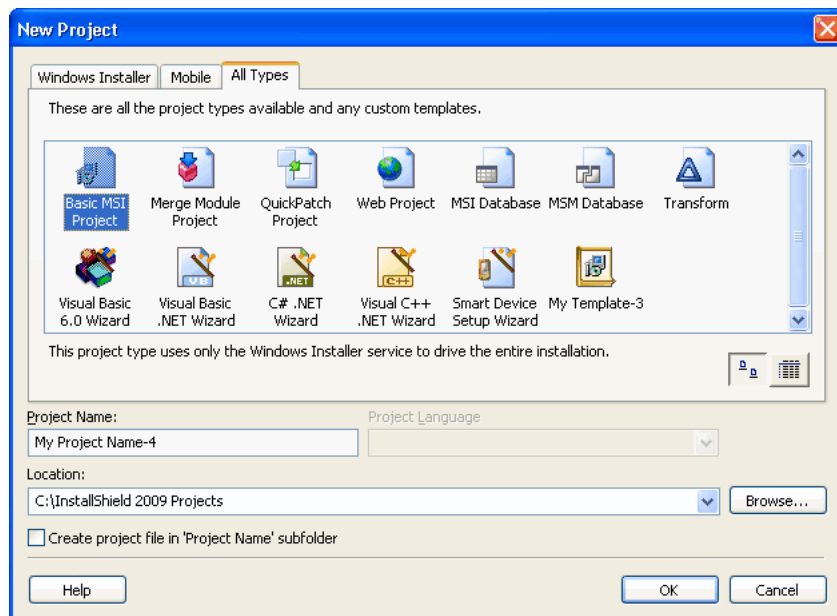
Creating an InstallShield Editor Template to Use Within Repackager



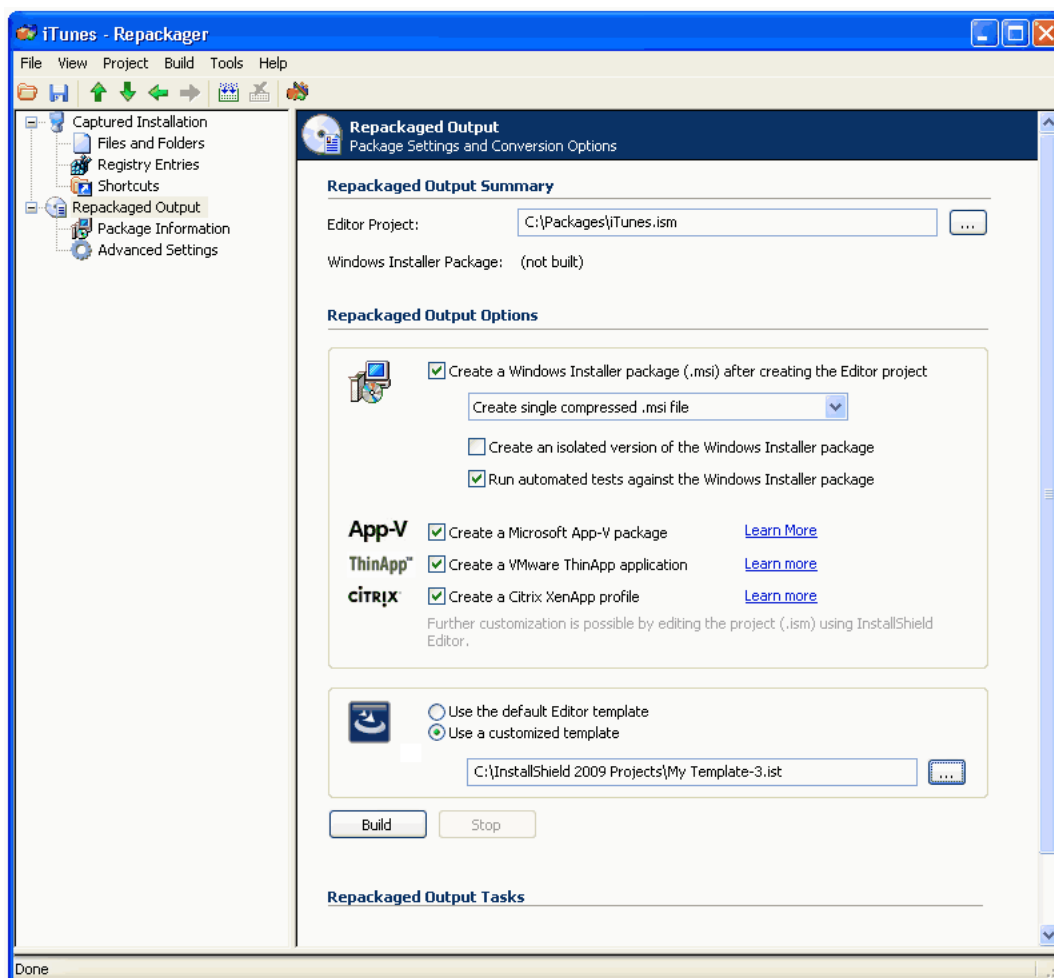
5. After making all required changes, save the project as an InstallShield Editor Template (.ist) type.



6. This new template should now be available along with other project types in the InstallShield Editor.



7. From within the Repackager interface, you can start using this customized template by selecting the **Use a customized template** option in the **Repackaged Output** view, and selecting the InstallShield Editor template that you just created.



Repackager Interface Reference

This section describes each of the dialog boxes and Wizard panels that you might encounter when using the Repackager interface. The help topics in this section are the same detailed documentation that is displayed when you press the F1 key or click the Help button while working in a dialog box.

- [Repackager Interface](#)
- [SmartScan Wizard](#)
- [Setup Intent Wizard](#)
- [VMware Repackaging Wizard](#)
- [Exclusions Editor Interface](#)
- [Options.ini File](#)

- [Files Associated with Repackager](#)
- [Using InstallShield to Chain Multiple Windows Installer Packages Together](#)
- [Troubleshooting](#)

Repackager Interface

From the Repackager Interface, you can:

- Open the Repackaging Wizard and repackage legacy setups.
- Open the Exclusions Editor and configure exclusions.
- Convert Novell ZENworks, Microsoft SMS, and WinINSTALL projects into Repackaging projects.
- Create a package exclusion list.
- Build a Repackager project into an InstallShield Editor project and Windows Installer package.

The Interface consists of several menus, a toolbar, the status bar, the output window, the View List, and several associated views.

- Menus and the toolbar are discussed in the [Menus and Toolbar](#) topic.
- Individual views are covered in their respective help topics.
- The status bar, output window, and View List are described in the following table.

Table 8-4 • Repackager Interface Elements

Interface Element	Description
Status Bar	The status bar, which can be toggled from the View menu, displays information when you hover over buttons in the toolbar.

Table 8-4 • Repackager Interface Elements

Interface Element	Description
View List	<p>The View List allows you to navigate to different views in the Repackager project. The corresponding view is displayed when you select an item in the tree. You can also use the Forward, Back, Navigate Up, and Navigate Down buttons in the View List.</p> <p>The View List includes the following views:</p> <ul style="list-style-type: none">• Captured Installation View• Files and Folders View• Registry Entries View• Shortcuts View• INI Files View• Deleted Files View• Deleted Registry Entries View• Repackaged Output View• Package Information View• Advanced Settings View
Output Window	<p>When you open Repackager 3.x output, Novell ZENworks projects, Microsoft SMS projects, WinINSTALL projects, or Wise installation projects in the Repackager Interface, conversion information appears in the Output window. This window can be toggled from the View menu.</p>

Repackager Start Page

When you first launch Repackager, the **Repackager Start Page** opens.

This page gives you a brief overview of Repackager functionality and uses, and gives you links to launch the Repackaging Wizard, convert a Windows Installer package to an application virtualization format, open an existing Repackager project, convert a legacy setup to a Repackager project, upgrade a legacy Repackager file, and open a recently accessed package.

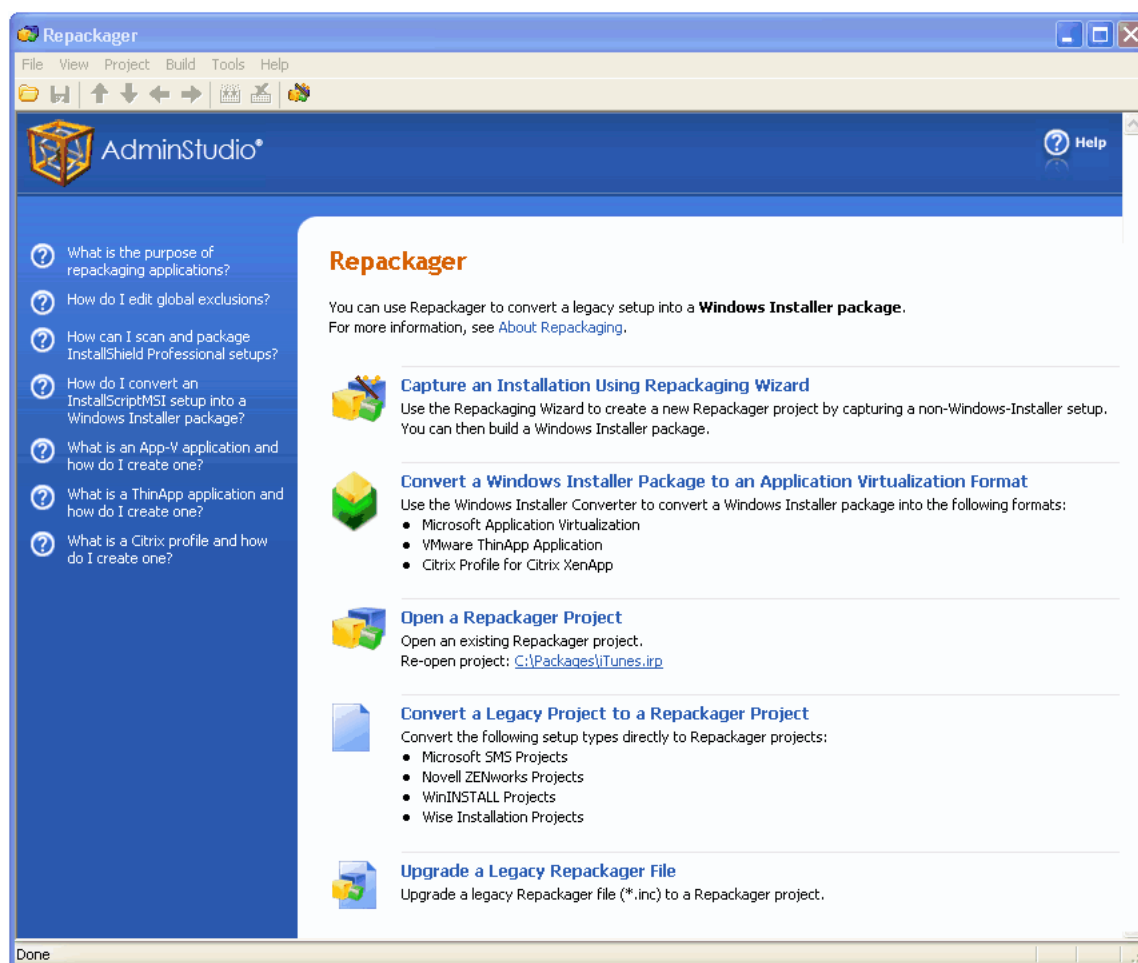


Figure 8-3: Repackager Start Page

Menus and Toolbar

The following table provides a description of each menu command and toolbar button:

Table 8-5 • Repackager Menus and Toolbars


Menu	Command	Toolbar Button	Keyboard Shortcuts	Description
File	Open		Ctrl+O	Allows you to open: <ul style="list-style-type: none"> • An existing Repackager project (.irp) • Repackager 3.x output (.inc) • Novell ZENworks 3.0, 3.1, or 4.0 project (.axt/.aot) • Microsoft SMS project (.ipf) • WinINSTALL converted project (.txt) (6.0, 6.5, or 7.x) • Wise Installer project (.wse)
File	Save		Ctrl+S	Saves the current project.
File	Save As			Saves the current project using the name and location you specify.
File	1,2,3,4			Allows you to open the four most recently accessed Repackager projects.
File	Exit			Exits Repackager.
View	Toolbar			Toggles display of the toolbar.
View	Status Bar			Toggles display of the status bar.
View	Output			Toggles display of the Output window.
View	Refresh		F5	Refreshes the current view.
Project	Edit Windows Installer Package			Once you build the Repackager project into a Windows Installer package (.msi), opens the package in InstallShield Editor (in Direct MSI Edit mode).
Project	Edit InstallShield Project			Once you build the Repackager project into an InstallShield Editor project (.ism), opens the project in InstallShield Editor.

Table 8-5 • Repackager Menus and Toolbars



Menu	Command	Toolbar Button	Keyboard Shortcuts	Description
Project	Setup Intent Wizard			Launches the Setup Intent Wizard.
Project	SmartScan Wizard			Launches the SmartScan Wizard , which allows you to scan an InstallShield Professional setup for additional files that may not have been captured during repackaging because they may be language- or platform-specific.
Project	Create Report		Ctrl+R	Allows you to create a report for the project in text or HTML format.
Project	Properties			Displays properties for the current project, including exclusion information.
Build	Build		F7	Builds the Repackager project into an InstallShield Editor project and a Windows Installer package.
Build	Stop Build		Ctrl+Break	Terminates an in-process build.
Tools	Repackaging Wizard			Launches the Repackaging Wizard.
Tools	Windows Installer Converter			Launches the Windows Installer Converter, which you can use to convert a Windows Installer package to a Microsoft App-V application, a ThinApp application, or a Citrix profile.
Tools	VMware Repackaging Wizard			Launches the VMware Repackaging Wizard.
Tools	Options			Displays the Options dialog box.
Tools	Isolation Options			Displays the Isolation Options dialog box, where you can specify assembly and digital signature isolation options.
Help	Contents			Launches the Help Library, displaying the Contents tab.

Table 8-5 • Repackager Menus and Toolbars

Menu	Command	Toolbar Button	Keyboard Shortcuts	Description
Help	Index			Launches the Help Library, displaying the Index tab.
Help	Search			Launches the Help Library, displaying the Search tab.
Help	Support Central			Accesses the AdminStudio Support Web site.
Help	Web Community			Accesses the AdminStudio Web Community.
Help	ReadMe			Displays the AdminStudio ReadMe file.
Help	Feedback			Accesses the feedback form on the AdminStudio Web site.
Help	AdminStudio on the Web			Accesses the AdminStudio Web site.
Help	About Repackager			Displays the About Repackager dialog box.
	Up			Moves you up one view in the View List.
	Down			Moves you down one view in the View List.
	Back			Displays the previously displayed view in the View List.
	Forward			Returns you to the view from which you selected the Back button.

Dialog Boxes

Repackager includes the following dialog boxes to assist you in your project creation:

- [Create Report Dialog Box](#)
- [Isolation Options Dialog Box](#)
- [Options Dialog Box](#)
- [Password Required Dialog Box](#)

- Project Properties Dialog Box
- WinINSTALL Conversion Dialog Box

About Repackager Dialog Box

This dialog box available by selecting **About Repackager** from the **Help** menu, displays version information for Repackager.

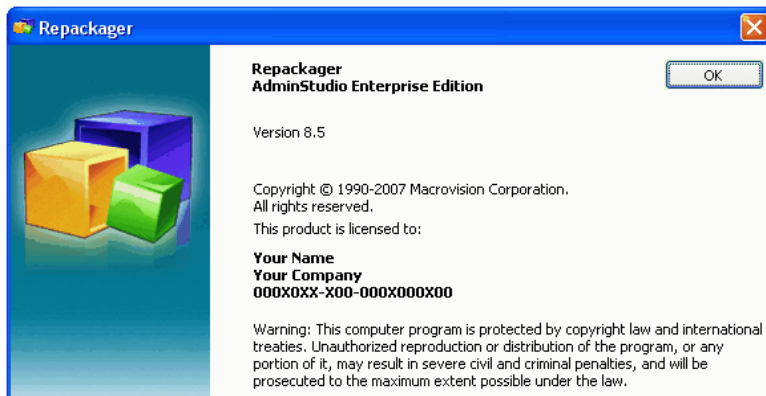


Figure 8-4: About Repackager Dialog Box

Create Report Dialog Box

The Create Report dialog box, available by selecting Create Report from the Project menu, allows you to configure a report for the current Repackager project, or a specific subset of captured data.

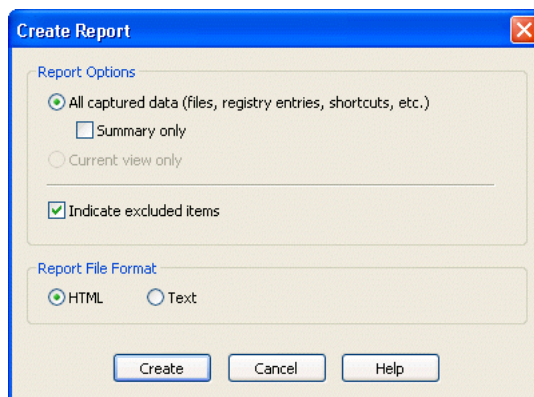


Figure 8-5: Create Report Dialog Box

This dialog box contains the following options:

Table 8-6 • Create Report Dialog Box Options

Option	Description
All captured data (files, registry entries, shortcuts, etc.)	Select to have the report include all captured data.
Summary only	If you select All captured data, you can select this option to only display summary information in the report (the number of items captured and the number of items excluded for files, .ini files, registry data, and shortcuts).
Current view only	Select this option to include only the currently selected view in the report.
Indicate excluded items	Select to display items that have been marked as excluded in Repackager.
Report File Format	Select the file format for Repackager reports: HTML or Text.
Create	When you click Create, you are prompted for a name and location for the outputted report.

Isolation Options Dialog Box

Application isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.

On the Isolation Options dialog box, which is opened by selecting **Isolation Options** from the **Tools** menu, you can specify the following Repackager isolation options:

- **Assembly Options**—Specify the type of assemblies Repackager will create, and the assembly naming conventions. See [Manifest Options Tab](#)
- **Digital Signature Options**—Configure the certificate information required when using shared assemblies. See [Digital Signature Tab](#).







Note • The modifications you make on the Isolation Options dialog box will be recorded in the *isolationconfig.ini* file, which is stored in the AdminStudio Shared directory.

Manifest Options Tab

The Manifest Options tab allows you to configure several settings associated with manifests. The following settings are included:

Table 8-7 • Isolation Options Dialog Box / Manifest Options Tab

Option	Description
Assembly Type	<p>This option allows you to select the type of assemblies that Repackager will create and use:</p> <ul style="list-style-type: none"> • Create private side-by-side assemblies in the application folder • Create shared side-by-side assemblies in the WinSxS folder (Default)  <p>Note • Manifests for shared assemblies must be digitally signed. This can be done in the Digital Signature Tab.</p>  <p>Note • A 2048-bit key is required to sign a Windows XP assembly/manifest being installed to the WinSxS folder.</p>
Assembly Naming Conventions	<p>Specify your company and division information to define the default naming convention that Repackager will use when creating assemblies during application isolation</p> <p>By default, assembly names are specified in the form of:</p> <p>Company.Division.Assembly</p>  <p>Note • See About Assemblies and About Manifests for more information.</p>
Create a new component for each assembly	<p>Select this option if you want to create a new component for each assembly created during isolation. This check box applies to all assemblies created.</p>  <p>Caution • If you are creating assemblies for applications files within multiple components, this option must be selected for successful application isolation.</p> <p>If you are planning to deploy this isolated package to operating systems prior to Windows XP, always check this box.</p>

Digital Signature Tab

On the **Digital Signature** tab, you can configure the certificate information required when using shared assemblies. This required digital signature provides an extra layer of protection, allowing you to obtain information about the company who created a global assembly.

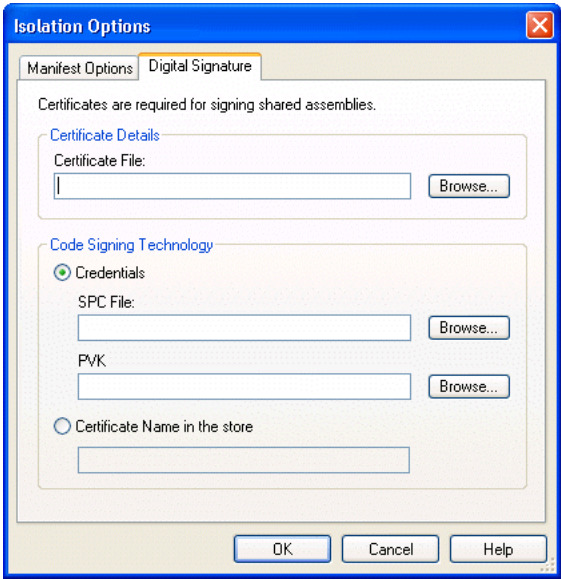


Figure 8-6: Isolation Options Digital Signature Options Tab



Caution • Repackager uses timestamping when signing global assemblies. Consequently, you must have an Internet connection on the computer when you create a global assembly.

You must configure the following options when signing these assemblies:

Table 8-8 • Isolation Options Dialog Box / Digital Signatures Tab



Item	Description
Certificate File	<p>Click the Browse () button next to the field and navigate to the certificate file you are using to sign assemblies.</p> <p>A digital certificate identifies you and/or your company to end users and assures them the data they are about to receive has not been altered.</p>
Credentials	<p>Select this option to use credential files as the code signing technology. If you select this option, you must supply the name and location of both your software publishing credential files: SPC File and PVK File.</p> <div></div> <p>Note • In order to receive a software publishing credentials and a private key, you must supply a certification authority, such as VeriSign, with specific information about your company and software.</p>
SPC File	Specify the name and location of your software publishing credentials file (. spc).
PVK	Specify the name and location of your private key file (. pvk).

Table 8-8 • Isolation Options Dialog Box / Digital Signatures Tab

Item	Description
Certificate Name in the Store	Select this option to use the name of an existing certificate file in the Certificate Store as the code signing technology. The Certificate Store is a central repository for certificate files. Using a Certificate Store allows you to reuse the certificate files for different purposes as necessary.



Note • For more information, see [About Digital Certificates](#).

Options Dialog Box

The Options dialog box, available from the **Tools** menu, presents options on three tabs: **Colors**, **Merge Modules**, and **Build Options**.

Colors Tab

On the **Colors** tab, you can configure the color of scanned items and deleted items in Repackager's exclusion views (**Files**, **.ini Files**, **Registry Data**, and **Shortcuts**).

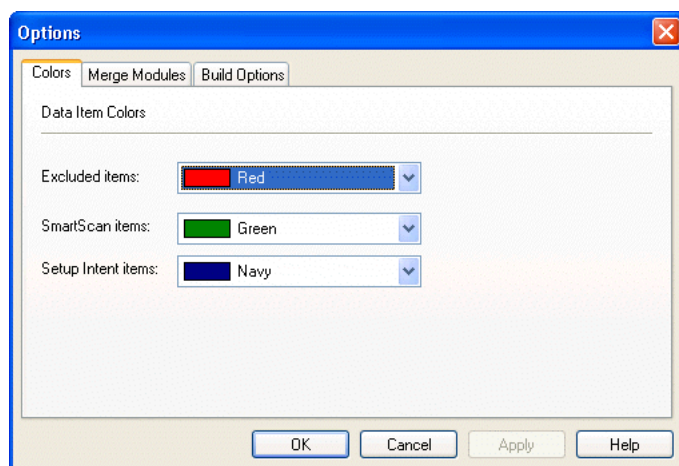


Figure 8-7: Colors Tab of the Options Dialog Box

Merge Modules Tab

On the **Merge Modules** tab, you can specify additional directories containing custom merge modules to use during repackaging.

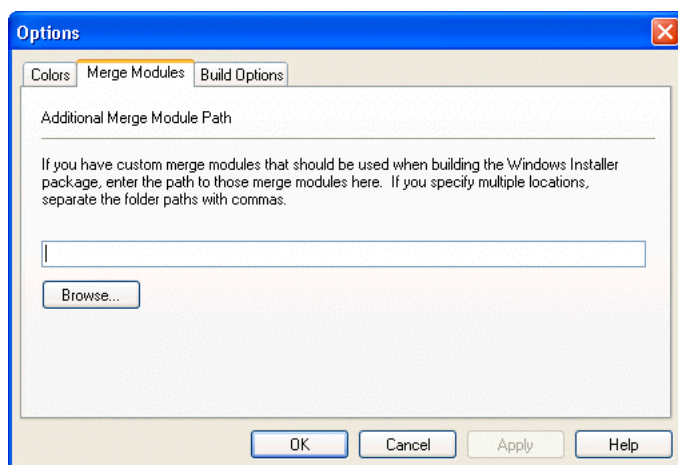


Figure 8-8: Merge Modules Tab of the Options Dialog Box

Build Options Tab

On the **Build Options** tab, you can specify whether or not you want to list ICE validation warnings in the Repackager output window during the Build process.

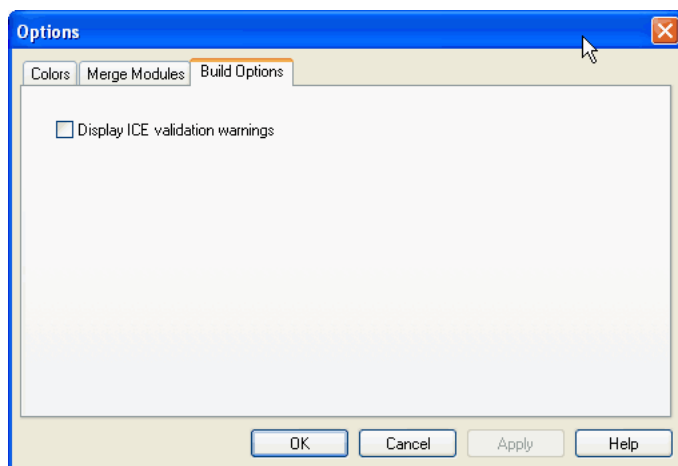


Figure 8-9: Build Options Tab of the Options Dialog Box

To display any ICE validation warnings that occur during the Repackager Build process, select the **Display ICE validation warnings** option. By default, this option is not selected.

Password Required Dialog Box

The Password Required dialog box is displayed when attempting to run the SmartScan Wizard on an InstallShield Professional setup which requires a password. Provide the password for the setup and click OK to proceed.

For InstallShield Professional 6.x and 7.x setups, the SmartScan Wizard can bypass password protection, and will not prompt you for a password even if one is required to run the installer.

Project Properties Dialog Box

The Project Properties dialog box, accessed by selecting Properties from the Projects menu, contains two tabs:

Table 8-9 • Project Properties Dialog Box Tabs

Tab	Description
General Tab	Allows you to view properties for the current Repackager project.
Exclusions Tab	Use to configure the location of the default exclusion file.

General Tab

The General tab of the Project Properties dialog box displays information about the current Repackager project (.irp).

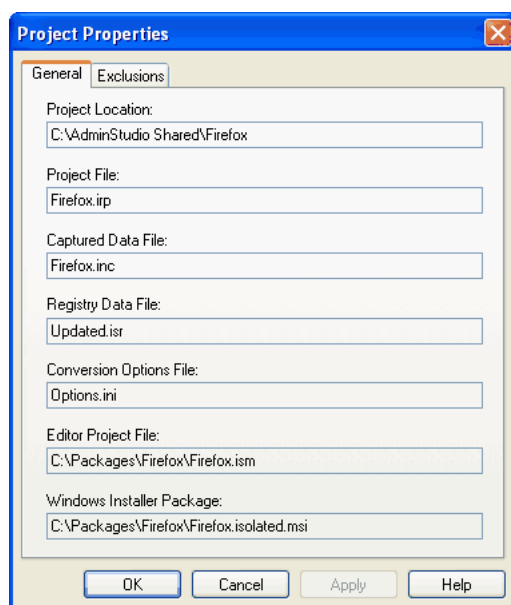


Figure 8-10: Project Properties Dialog Box General Tab

The following options are displayed:

Table 8-10 • General Tab Options

Option	Description
Project Location	The full path of the current Repackager project file (.irp).
Project File	The name of the current Repackager project file.

Table 8-10 • General Tab Options

Option	Description
Captured Data File	The name and location of the captured data file (.inc), which was either created by the Repackaging Wizard or during conversion of a Novell ZENworks project, Microsoft SMS project, or WinINSTALL project. The path is relative to the current Repackager project file.
Registry Data File	The name and location of the file containing captured registry data. The path is relative to the current Repackager project file.
Conversion Options File	The name and location of the Options.ini file, which contains an exhaustive list of all options you can use during conversion of the Repackager project to an InstallShield Editor project and Windows Installer package.
Editor Project File	The name and location of the InstallShield Editor project file as set in the Product View (MSI Package). The path is relative to the current Repackager project file.
Windows Installer Package	The name and location of the Windows Installer package. The path is relative to the current Repackager project file.

Exclusions Tab

The Exclusions tab allows you to select an exclusion file to use as a filter when importing captured data into a Repackager project.

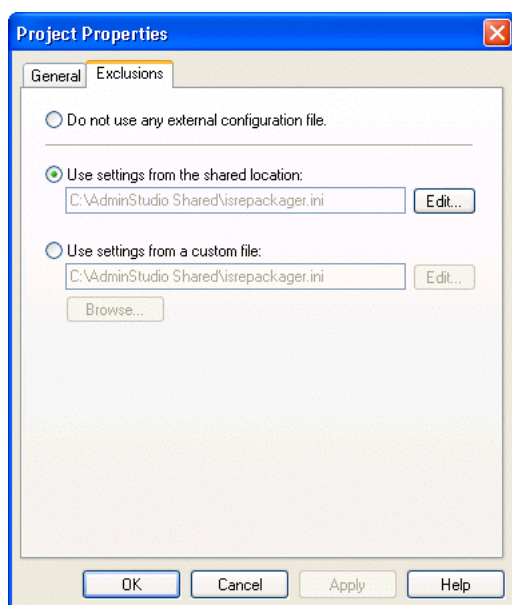


Figure 8-11: Project Properties Dialog Box Exclusions Tab

Select one of the following options for the configuration file:

Table 8-11 • Exclusions Tab Properties

Option	Description
Do not use any external configuration	Repackager will import all captured data into the Repackager project.
Use settings from the shared location	Repackager will use the settings contained in isRepackager.ini in the AdminStudio Shared directory (configured during installation). Use this option when you are working in a team environment where the exclusion list needs to be stored in a centralized location.
Use settings from InstallShield defaults	Repackager will use the settings contained in the default.ini file in the Repackager folder. These are the InstallShield Editor-recommended exclusions. It is recommended that you do not modify these exclusions so you can return to them if you need to restart your exclusion list.
Edit	Click to open the Exclusions Editor, which you can use to exclude files, registry entries, .ini files, or shortcuts from the Repackager project. See Configuring Exclusions Using the Exclusions Editor and Exclusions Editor Interface for more information.
Use settings from a custom file	Specify or browse to a file created with the Exclusions Editor that you want to use as your filter during conversion to a Repackager project. You would create a custom exclusion file based upon your company's requirements.



Caution • Using the custom settings option, it is possible to use the local settings file (*isRepackager.ini*) in the Windows directory. This file is also used for default exclusions for the Repackaging Wizard. By modifying this file, you introduce the possibility of excluding data at repackaging time in subsequent Repackaging Wizard executions, as opposed to marking items as excluded in a Repackager project (which does not affect the captured data). For this reason, it is highly recommended that you do not use the *isRepackager.ini* configuration file in the local Windows folder for your Repackager exclusions.

WinINSTALL Conversion Dialog Box

When you convert a WinINSTALL project to a Repackager project, this dialog box appears to allow you to set WinINSTALL-specific variables. These variables are:

Table 8-12 • WinINSTALL Variables

Variable	Description
@Server	The machine name of the server where the WinINSTALL directory is located.

Table 8-12 • WinINSTALL Variables

Variable	Description
@WinstallDir	The location of the directory where the WinINSTALL executables are located.

Repackager Views

Repackager includes several views, from which you can examine the captured data that will be used to create an InstallShield Editor project (.ism) and Windows Installer package (.msi). Depending on the presence or absence of certain data types, some views may not be displayed. For example, if the setup does not include any .ini files, the INI Files view will not be displayed in the View List.

The following views are available in Repackager:

- [Captured Installation View](#)
- [Files and Folders View](#)
- [Registry Entries View](#)
- [Shortcuts View](#)
- [INI Files View](#)
- [Deleted Files View](#)
- [Deleted Registry Entries View](#)
- [Repackaged Output View](#)
- [Package Information View](#)
- [Advanced Settings View](#)



Note • Information listed in the views (such as files, .ini files, or registry entries) is limited to 267 characters in length. Anything longer than this limit will be truncated in the view. The full value can be viewed in InstallShield Editor.

Captured Installation View

From the **Captured Installation** view, you can review summary information about the setup you are converting into a Windows Installer package.

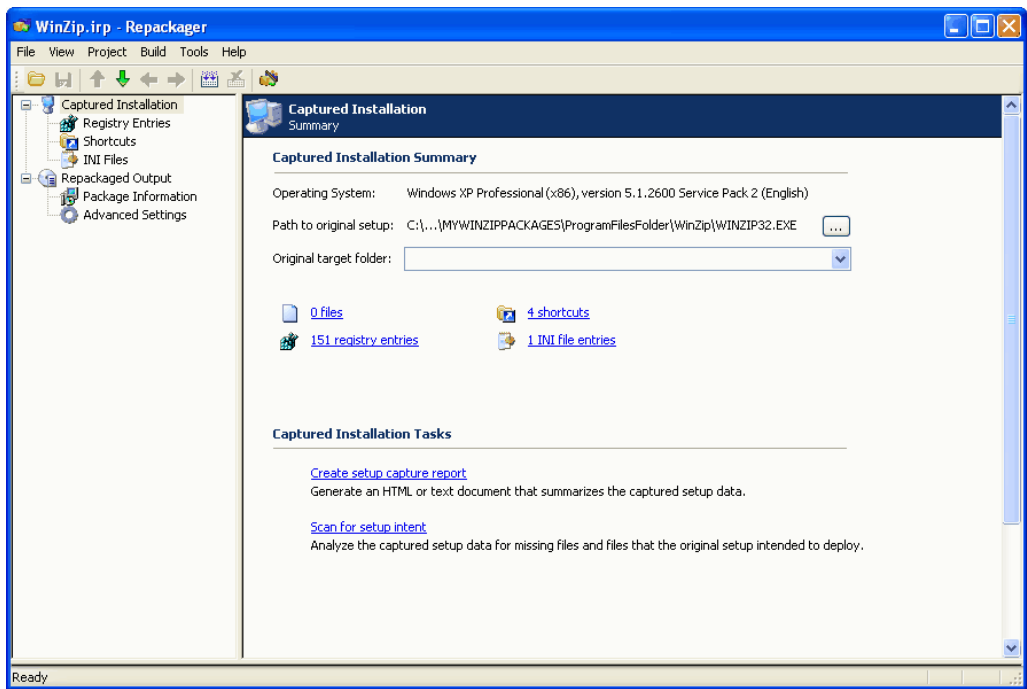


Figure 8-12: Repackager Captured Installation View

The **Captured Installation** view lists the following information:

Table 8-13 • Repackager Captured Installation View





Property	Description
Operating System	Identifies the operating system—including version, service pack, and processor type (32 or 64-bit)—of the machine where the capture was performed.
Path to original setup	Location of setup that was repackaged. Click the browse () button to select a different setup.
Original target folder	<p>From this list, select the original target folder for the installation. In most cases, this will be a subdirectory of [ProgramFilesFolder].</p> <p>Alternatively, you can enter your own target. This value will be set as the value for INSTALLDIR, and is a mandatory property.</p> <div></div> <p>Note • Information about the provided install locations can be found in the SystemFolder Property topic of the Windows Installer Help Library.</p>

Table 8-13 • Repackager Captured Installation View

Property	Description
Number of Files, Shortcuts, Registry Entries, and INI File Entries	<p>Links that list the number of files, shortcuts, and registry entries captured, and the number of .ini file changes made. Click these links to open the following subviews:</p> <ul style="list-style-type: none"> • Files and Folders View • Registry Entries View • Shortcuts View • INI Files View <p>Each subview of this view allows you to view the names and associated information of each item captured, and selectively exclude (or reinclude) these items from the ultimate Windows Installer package.</p> <p>If no entries were captured of a particular type, the corresponding view does not appear in the View List. For example, if no .ini file changes were captured, the INI Files view is not displayed.</p>
Create setup capture report	<p>Click to generate the Setup Capture Report, an HTML or text document that summarizes the data that was captured when a setup was repackaged. For more information, see Creating a Setup Capture Report for a Project.</p>  <hr/> <p>Edition • The Setup Capture Report feature is included with AdminStudio Standard, Professional, and Enterprise Editions.</p>
Scan for setup intent	<p>Click to launch the Setup Intent Wizard, which you can use to scan a setup to identify files that may not have been captured during repackaging—effectively recognizing the installation’s intent for these files. For more information, see Using the Setup Intent Wizard to Detect File Dependencies in a Repackager Project.</p>  <hr/> <p>Edition • The Setup Intent Wizard is included with AdminStudio Standard, Professional, and Enterprise Editions.</p>

Automatic Launch of the SmartScan Wizard

If you are repackaging a legacy installation that was originally created using InstallShield Professional 5.5 or later, you can use the **SmartScan Wizard** to scan the setup for possible additional files, .ini files, shortcuts, or registry data (such as for language-specific setups or platform-specific setups). The **SmartScan Wizard** is able to group files, shortcuts and registry entries into features corresponding to InstallShield Professional components. Any items (files, folders, shortcuts, or registry entries) that are attached to a component directly or indirectly (through File Groups), are attached to their corresponding features.

For your convenience, if a Repackaging project is opened (either by using **Open** on the **File** menu or by using the Repackaging Wizard), and the project meets the following conditions, the SmartScan Wizard is automatically launched if:

- the original setup was an InstallShield Professional 5.5 or later installation, *and*
- the file has not already been repackaged using the InstallShield Professional Logging Method (available for InstallShield Editor and DevStudio 9.x InstallScript installations only), *and*
- the SmartScan Wizard has not already been run on this project.

Files and Folders View

From the **Files and Folders** view, you can examine information about each captured file, selectively exclude files or directories from the package you are creating, or reinclude files that you previously excluded.

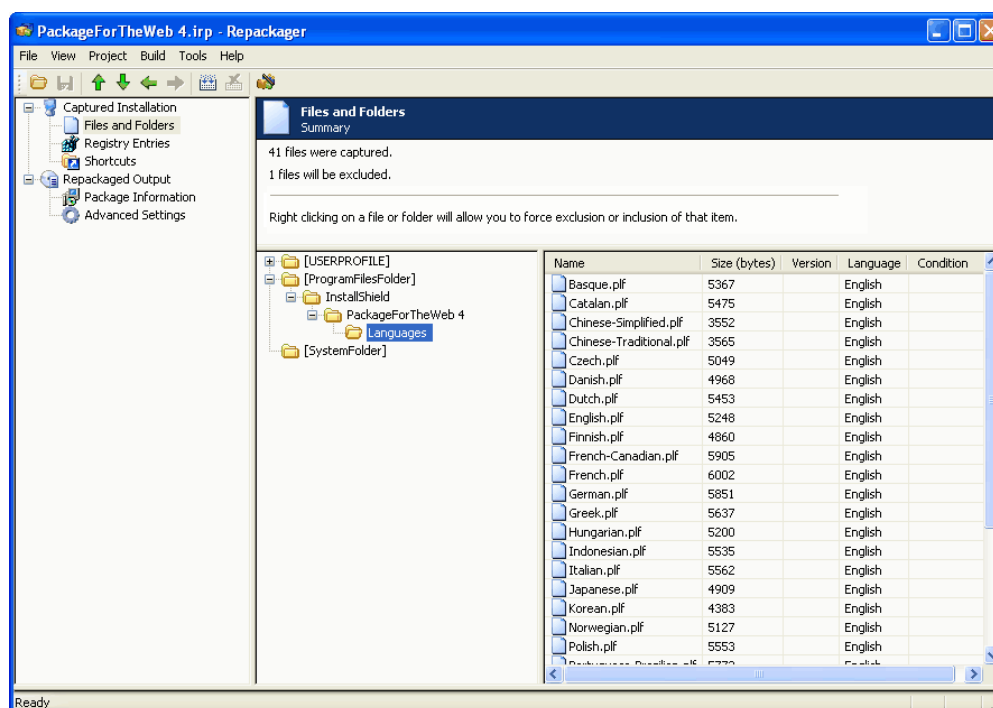


Figure 8-13: Repackager Files and Folders View

The upper pane displays the number of files captured and how many of these files will be excluded from the Windows Installer package when built. The lower-left pane provides a tree from which you can see where files will be installed and the names of the files.

When you select a file from the tree, the lower-right pane displays attributes for that file. These attributes are:

Table 8-14 • File Attributes

Attribute	Description
Name	The file's name.
Size	The file's size in bytes.

Table 8-14 • File Attributes (cont.)

Attribute	Description
Version	The file's version.
Short Name	The short name for the file (if the file's author defined it).
Language	The file's language.

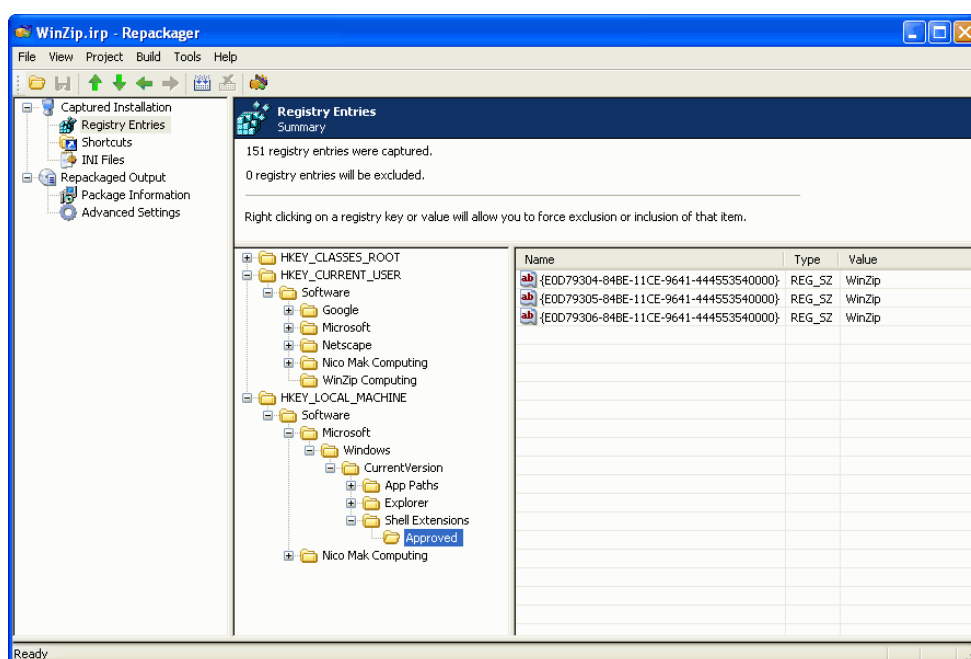
Excluding Files and Subdirectories

To specify which files and subdirectories you want to include in the package, use the **Exclude**, **Exclude All**, **Include**, and **Include All** commands on the context menu:

- **To exclude a captured file**, select the file you want to exclude and select **Exclude**.
- **To exclude captured files within a directory**, select the directory containing the files you want to exclude and select either **Exclude** (to exclude only the files in the selected directory) or **Exclude All** (to exclude all of the files in the selected directory and all of its subdirectories).
- **To include a captured file that had previously been excluded**, select the file you want to include and select **Include**.
- **To include captured files within a directory that had previously been excluded**, select the directory containing the files you want to include and select either **Include** (to include only the files in the selected directory) or **Include All** (to include all of the files in the selected directory and all of its subdirectories).

Registry Entries View

From the **Registry Entries** view, you can examine information about each captured registry entry, selectively exclude registry values or registry keys from the package you are creating, or reinclude registry values that you previously excluded.



The upper pane displays the number of registry entries captured and how many of these entries will be excluded from the Windows Installer package when built. The lower-left pane provides a tree displaying the registry keys and subkeys captured. When you select a key from the tree, the lower-right pane displays any registry values for that key. Displayed information includes:

Table 8-15 • Registry Attributes

Attribute	Description
Name	The registry value name.
Type	The registry value type. This can be either a string value, an expandable string value, a multistring value, a dword value, or a binary value.
Value	The content of the registry value.

Excluding Registry Entries

To specify which registry entries you want to include in the package, use the **Exclude**, **Exclude All**, **Include**, and **Include All** commands on the context menu:

- **To exclude a registry entry**, select the registry entry you want to exclude and select **Exclude**.
- **To exclude registry entries within a registry key or registry hive**, select the key or hive from the tree and select either **Exclude** (to exclude the registry entries in the selected hive or key only) or **Exclude All** (to exclude all of the registry entries in the selected hive or key and all of its keys and subkeys).
- **To include a registry entry that had previously been excluded**, select the registry entry and select **Include**.

- To include registry entries within a registry key or registry hive that had previously been excluded, select the key or hive from the tree and select either **Include** (to include the registry entries in the selected hive or key only) or **Include All** (to include all of the registry entries in the selected hive or key and all of its keys and subkeys).

Shortcuts View

From the Shortcuts view, you can examine information about each captured shortcut, selectively exclude shortcuts from the package you are creating, or reinclude shortcuts that you previously excluded.

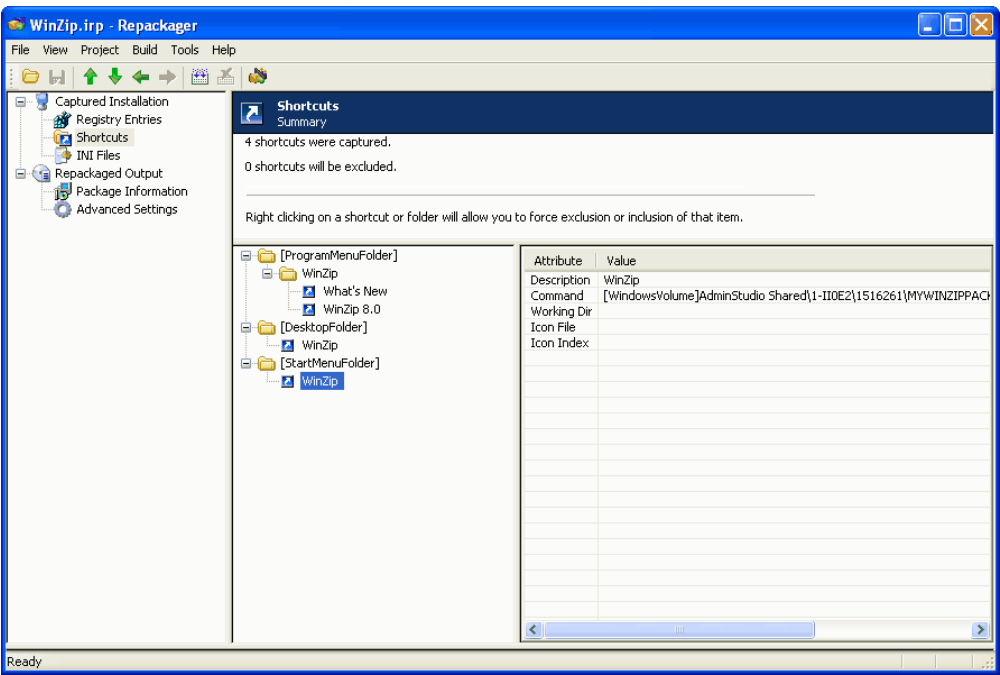


Figure 8-14: Repackager Shortcuts View

The upper pane displays the number of shortcuts captured and how many of these shortcuts will be excluded from the Windows Installer package when built. The lower-left pane provides a tree from which you can see where shortcuts will be installed and the names of the shortcuts. When you select a shortcut from the tree, the lower-right pane displays attributes for that shortcut. These attributes are:

Table 8-16 • Shortcuts View Attributes

Attribute	Description
Description	The name of the shortcut as it appears on the desktop.
Command	The fully-qualified path and name of the file to which the shortcut points.

Table 8-16 • Shortcuts View Attributes

Attribute	Description
Working Dir	The shortcut's working directory, which may need to be specified so required files can load. This is equivalent of the Start in value found when right-clicking a shortcut from the desktop and selecting Properties.
Icon File	The name of the file containing the shortcut's icon.
Icon Index	The index number for the icon in the icon file.



Note • Shortcuts can be excluded from the Windows Installer package you are building on an individual shortcut basis or by directory.

Excluding Shortcuts

To specify which shortcuts you want to include in the package, use the **Exclude**, **Exclude All**, **Include**, and **Include All** commands on the context menu:

- **To exclude a shortcut**, select the shortcut you want to exclude and select **Exclude**.
- **To exclude shortcuts within a directory**, select the directory containing the shortcuts you want to exclude and select either **Exclude** (to exclude only the shortcuts in the selected directory) or **Exclude All** (to exclude all of the shortcuts in the selected directory and all of its subdirectories).
- **To include a shortcut that had previously been excluded**, select the shortcut you want to include and select **Include**.
- **To include shortcuts within a directory that had previously been excluded**, select the directory containing the shortcuts you want to include and select either **Include** (to include only the shortcut in the selected directory) or **Include All** (to include all of the shortcuts in the selected directory and all of its subdirectories).

INI Files View

From the INI Files view, you can examine information about each captured .ini file, selectively exclude .ini files or .ini file sections from the package you are creating, or reinclude .ini files or sections that you previously excluded.

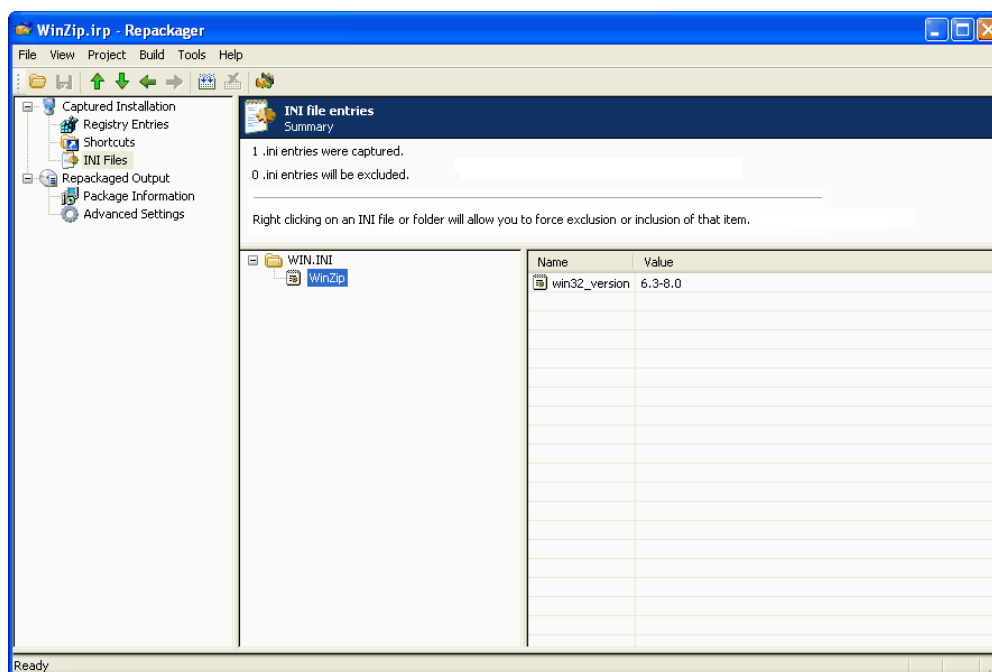


Figure 8-15: Repackager INI Files View

The upper pane displays the number of .ini files captured and how many of these .ini files will be excluded from the Windows Installer package when built. The lower-left pane provides a tree from which you can see the full path to captured .ini files and sections contained within the .ini files. When you select a section from the tree, the lower-right pane displays name/value pairs in that section.

Excluding INI Files

To specify which INI files you want to include in the package, use the **Exclude**, **Exclude All**, **Include**, and **Include All** commands on the context menu:

- **To exclude an INI file**, select the INI file you want to exclude and select **Exclude**.
- **To exclude INI files within a directory**, select the directory containing the INI files you want to exclude and select either **Exclude** (to exclude only the INI files in the selected directory) or **Exclude All** (to exclude all of the INI files in the selected directory and all of its subdirectories).
- **To include an INI file that had previously been excluded**, select the INI file you want to include and select **Include**.
- **To include INI files within a directory that had previously been excluded**, select the directory containing the INI files you want to include and select either **Include** (to include only the INI file in the selected directory) or **Include All** (to include all of the INI files in the selected directory and all of its subdirectories).

Deleted Files View

From the Deleted Files view, you can examine information about each file deleted during repackaging, selectively exclude files or directories from the package you are creating, or reinclude previously excluded files.

The upper pane displays the number of files captured and how many of these files will be excluded from the Windows Installer package when built. The lower-left pane provides a tree from which you can see where files will be installed and the names of the files. When you select a file from the tree, the lower-right pane displays attributes for that file. These attributes are:

Table 8-17 • Deleted Files View Attributes

Attribute	Description
Name	The file's name.
Size	The file's size in bytes.
Version	The file's version.
Short Name	The short name for the file (if the file's author defined it).
Language	The file's language.

Excluding Files and Subdirectories

To specify which files and subdirectories you want to include in the package, use the Exclude, Exclude All, Include, and Include All buttons:

- To exclude a captured file from the package, select the file you want to exclude and click Exclude.
- To exclude all captured files and subdirectories within a directory from the package, select the directory containing the files and subdirectories you want to exclude and click Exclude All.
- To include a captured file in the package that had previously been excluded, select the file you want to include and click Include.
- To include all captured files and subdirectories within a directory, select the directory containing the files and subdirectories you want to include and click Include All.

Deleted Registry Entries View

From the Deleted Registry Entries view, you can examine information deleted from the registry repackaging, selectively exclude registry keys from the package you are creating, or reinclude previously excluded data.

The upper pane displays the number of deleted registry entries captured and how many of these entries will be excluded from the Windows Installer package when built. The lower-left pane provides a tree displaying the registry keys and subkeys captured. When you select a key from the tree, the lower-right pane displays any registry values for that key. Displayed information includes:

Table 8-18 • Deleted Registry Entries View Attributes

Attribute	Description
Name	The registry value name.

Table 8-18 • Deleted Registry Entries View Attributes

Attribute	Description
Type	The registry value type. This can be either a string value, an expandable string value, a multistring value, a dword value, or a binary value.
Value	The content of the registry value.

Excluding Registry Entries

To specify which registry entries you want to include in the package, use the Exclude, Exclude All, Include, and Include All buttons:

- To exclude a registry entry from the package, select the registry entry you want to exclude and click Exclude.
- To exclude all registry entries and subdirectories within a directory from the package, select the directory containing the registry entries you want to exclude and click Exclude All.
- To include a registry entry in the package that had previously been excluded, select the registry entry you want to include and click Include.
- To include all registry entries and subdirectories within a directory, select the directory containing the shortcuts and subdirectories you want to include and click Include All.

Repackaged Output View

From this view, you can configure build options for the project, including whether to build an MSI package automatically following conversion.

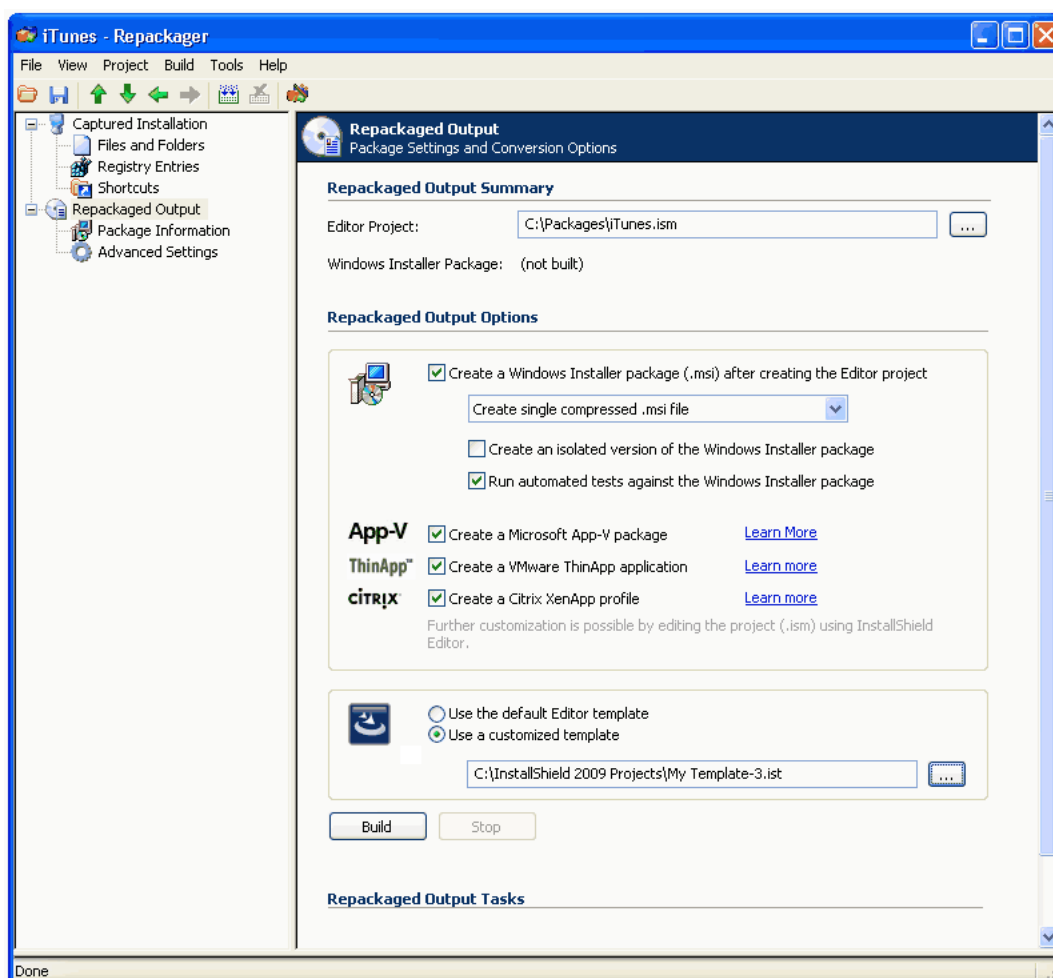


Figure 8-16: Repackager Repackaged Output View

The following properties are available for configuration:

Table 8-19 • Repackaged Output View Options

Option	Description
Editor project	Provide the name and location of the InstallShield Editor project (.ism) file.
Windows Installer package	The name and location of the Windows Installer package (.msi). If a Windows Installer package has not yet been built from this Repackager project, (not built) is listed.
Create a Windows Installer package (.msi) after creating the Editor project	If this option is selected, after creating the InstallShield Editor project file (.ism), a Windows Installer (.msi) file will also be built.

Table 8-19 • Repackaged Output View Options

Option	Description
Windows Installer Package Options	<p>If you have selected the Create a Windows Installer package after creating the Editor project option, you need to also select one of the following options:</p> <ul style="list-style-type: none">• Create single, compressed .msi file—Select this option if you want to compress all necessary files inside the .msi package, as opposed to storing them outside of the .msi database.• Create single, compressed setup.exe file—Select this option if you want to compress all files inside a setup.exe file, including the .msi file and all other necessary files.• Create .msi file + external compressed .cab file—Select this option if you want to create an .msi file and want to compress the rest of the necessary files in an external .cab file.• Create .msi file + external compressed .cab file + setup.exe—Select this option if you want to create an .msi file and a setup.exe file, and want to compress all the rest of the necessary files in an external .cab file.• Create uncompressed .msi file—Select this option if you want to create an uncompressed .msi file. All of the rest of the necessary files, in uncompressed format, would be shipped with the .msi file.• Create uncompressed .msi file + setup.exe—Select this option if you want to create an uncompressed .msi file along with a setup.exe file. All of the rest of the necessary files, in uncompressed format, would be shipped with the .msi and setup.exe files.

Table 8-19 • Repackaged Output View Options






Option	Description
Create an isolated version of the Windows Installer package	<p>Select this option to create a second, isolated version of the Windows Installer package when the Windows Installer package is built.</p> <p>Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.</p> <p>If this option is selected, an additional Windows Installer package will be created in the same directory as the .ism file and the other .msi file, with the naming convention of:</p> <p><i>appname.isolated.msi</i></p> <p>For more information on how Repackager isolates applications and the available isolation options, see Isolating Windows Installer Packages.</p>  <p>Note • This option is only enabled when the Create a Windows Installer package (.msi) after creating the Editor project option is selected and one of the following values is chosen:</p> <ul style="list-style-type: none"> • Create single compressed .msi file • Create .msi file + external compressed .cab file • Create uncompressed .msi file
Run automated tests against the Windows Installer package	<p>Select this option to automatically run PackageExpert tests against the newly built Windows Installer package to determine if it is built according to Windows Installer standards, and if it is in compliance with the installation requirements of the Windows Vista operating system.</p> <p>All of the tests that are currently selected on the PackageExpert Configuration View are run, and those selected tests that are automatically resolvable, will be resolved.</p>  <p>Note • This option is only enabled when the Create a Windows Installer package (.msi) after creating the Editor project option is selected and any of the values except for Create single compressed setup.exe file is chosen.</p>
Create a Microsoft App-V application	<p>If this option is selected, after building a Windows Installer (.msi) file, a Microsoft App-V application will also be built.</p>  <p>Note • This option requires that you build a Windows Installer package.</p>

Table 8-19 • Repackaged Output View Options

Option	Description
Create a VMware ThinApp application	<p>If this option is selected, after building a Windows Installer (.msi) file, a VMware ThinApp application will also be built.</p>  <p>Note • This option requires that you build a Windows Installer package.</p>
Create a Citrix XenApp profile	<p>If this option is selected, after building a Windows Installer (.msi) file, a Citrix profile compatible with Citrix XenApp will also be built.</p>  <p>Note • This option requires that you build a Windows Installer package.</p>
Use the default Editor template	<p>When building an InstallShield Editor project, select this option to use the default InstallShield Editor template.</p> <p>A project template contains all of the default settings and design elements that you want to use as a starting point when you create an installation project.</p>
Use a customized template	<p>When building an InstallShield Editor project, select this option to specify a customized InstallShield Editor Project Template to use.</p> <p>For example, if you wanted all of your InstallShield Editor projects to have a special custom dialog, a set of required redistributables, and a particular SQL script, you could create a project template that has all of those settings. Then, any time that you wanted to create a new project, you could base it off of your custom template. This enables you to avoid re-creating the custom dialog, re-adding the redistributables, and re-adding the SQL script every time that you create a new InstallShield Editor Project.</p>
Build	<p>Click to initiate the build process to build a Windows Installer package.</p>
Repackaged Output Tasks	<p>After an InstallShield Editor project and a Windows Installer package has been built, you can use these links to perform the following tasks:</p> <ul style="list-style-type: none"> • Modify the Editor Project—Open this Repackager project's associated InstallShield Editor project in InstallShield Editor. • Modify the Windows Installer package with Editor—Open this Repackager project's associated Windows Installer package in InstallShield Editor.

Once you have built the Windows Installer package and/or InstallShield Editor file, you can launch InstallShield Editor from the **Repackaged Output** area of the view.

Package Information View

The Package Information view allows you to specify information for the Windows Installer package that you build from the Repackager project. Much of this information may be prepopulated based on settings used in the Repackaging Wizard.

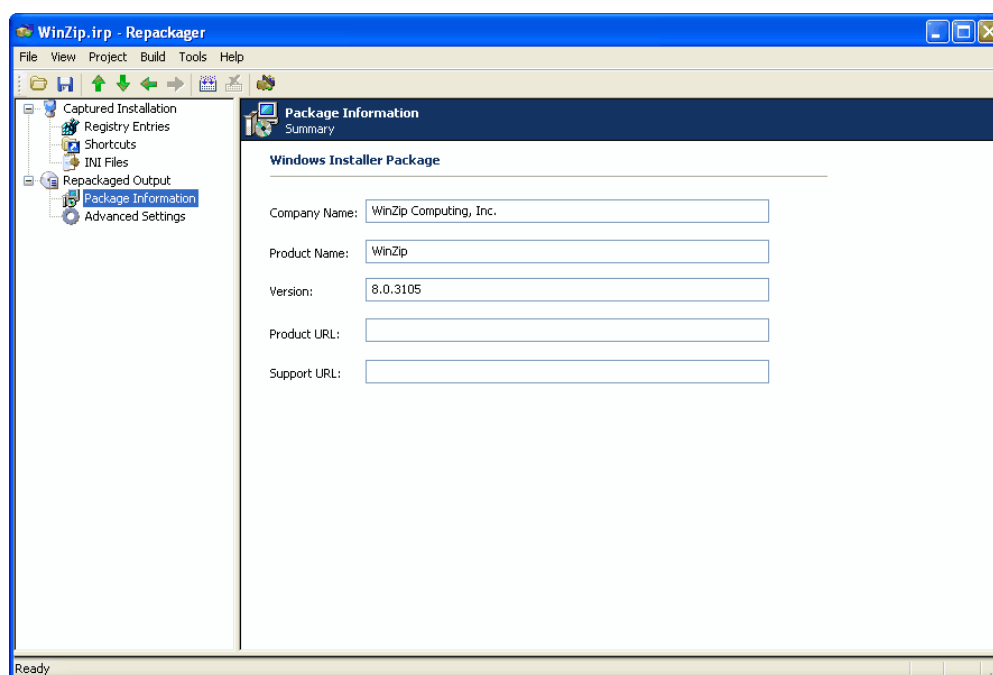


Figure 8-17: Repackager Package Information View

You can configure the following options:

Table 8-20 • Package Information View Options

Option	Description
Company Name	The name of the company that developed the product you are repackaging.
Product Name	The name of the product you are repackaging.
Version	The product's version number.
Product URL	The URL for product information. This appears in Add/Remove Programs in the Control Panel.
Support URL	A URL for support information. This also appears in Add/Remove Programs in the Control Panel, and is often changed during repackaging to provide an internal support URL.

Advanced Settings View

From the Advanced Settings view, you can configure several additional settings that may apply to your repackaged setup.

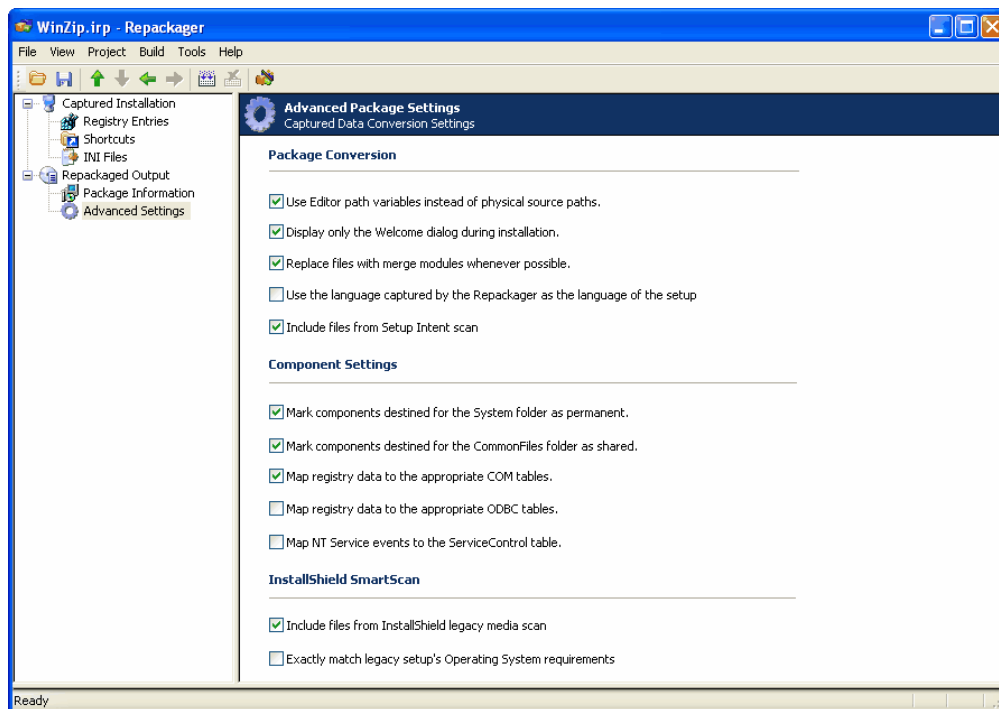


Figure 8-18: Repackager Advanced Settings View

Select the appropriate options:

Package Conversion Options

The following package conversion options are available:

Table 8-21 • Package Conversion Options

Option	Description
Use Editor path variables instead of physical source paths	When storing files in the InstallShield Editor project (.ism), the Wizard uses path variable locations whenever possible.
Display only the Welcome dialog box during installation	Only the Welcome dialog box is displayed when the Windows Installer package is run on a target machine. If this option is unchecked, the default UI sequence is displayed when the setup is installed.
Replace files with merge modules wherever possible	Following best practice rules, Repackager replaces components with comparable merge modules whenever possible.

Table 8-21 • Package Conversion Options (cont.)

Option	Description
Use the language captured by the Repackager as the language of the setup	When selected, the target package's language will be the language detected by Repackager (as displayed in the Captured Installation view).
Include files from Setup Intent scan	Any files identified when running the Setup Intent Wizard will be included in the package (unless you have manually excluded them from the project).

Component Settings Options

The following component settings options are available:

Table 8-22 • Component Settings Options

Option	Description
Mark components destined for the System folder as permanent	Executable files installed to the system folder (System32Folder) are marked as Permanent files and will not be uninstalled when the package is uninstalled. This eliminates ICE09 validation errors.
Mark components destined for the CommonFiles folder as shared	Executable files installed to the CommonFilesFolder (or a subfolder of CommonFilesFolder) are marked as shared files. This ensures that these components can coexist with DLLs installed by previous setups.
Map registry data to the appropriate COM tables	Setting this option reduces the number of ICE33 warnings that can occur during package validation, resulting from data not being mapped to the appropriate MSI tables.
Map registry data to the appropriate ODBC tables	If selected, ODBC-related registry data is mapped to ODBC tables instead of the Registry table. This data will only function correctly if Windows Installer supports the ODBC resource being mapped; it is highly recommended that you do not enable this option if you are unsure whether the ODBC resources are supported correctly by Windows Installer.
Map NT Service events to the ServiceControl table	If selected, NT Service-related registry data is mapped to ServiceControl table instead of the Registry table.

InstallShield SmartScan Options

The following InstallShield SmartScan options are available:

Table 8-23 • InstallShield SmartScan Options

Option	Description
Include files from InstallShield legacy media scan	Files identified in the Media Scan Wizard will be included in the package (unless you have manually excluded them from the project).
Exactly match legacy setup's Operating System requirements	For SmartScan projects or projects using the Professional Logging Method: <ul style="list-style-type: none">• Selected—if this option is selected, component conditions will store specific operating system information: if the filter was NT4, the condition will be (VersionNT=4).• Unselected—If this option is not selected, component conditions will store a grouping of the operating system: if the filter was NT4, the condition will be (VersionNT).• Default—By default, this option is not selected.

SmartScan Wizard

The SmartScan Wizard is designed to scan original, InstallShield Professional 5.x and later media for setup information that may not have been captured during repackaging, such as possible additional files, .ini files, shortcuts, or registry data, which is often included in language-specific or platform-specific setups. This primarily involves OS-dependent and language-dependent files that may be included in header files. Ultimately, this makes the installation more portable.



Tip • Any files found will be displayed in Repackager in a different color (as specified in the Color tab of the Options dialog box).

The SmartScan Wizard is also able to group files, shortcuts and registry entries into features corresponding to InstallShield Professional components. Any items (files, folders, shortcuts, or registry entries) that are attached to a component directly or indirectly (through File Groups), are attached to their corresponding features.

Automatic Launch of the SmartScan Wizard

For your convenience, if a Repackaging project is opened (either by using Open on the File menu or by using the Repackaging Wizard), and the project meets certain conditions, the SmartScan Wizard is automatically launched. The SmartScan Wizard is launched:

- if the original setup was an InstallShield Professional 5.5 or later installation, and
- the file has not already been repackaged using the [InstallShield Professional Logging Method](#) (available for InstallShield Editor and DevStudio 9.x InstallScript installations only), and

- the SmartScan Wizard has not already been run on this project

SmartScan Wizard Panels

The SmartScan Wizard includes the following panels:

- [Welcome to the SmartScan Wizard Panel](#)
- [Original InstallShield Professional Setup Panel](#)
- [Scanning Panel](#)
- [Setup Feature Tree Panel](#)
- [Scanning Media Panel](#)

Welcome to the SmartScan Wizard Panel

The SmartScan Wizard is designed to scan original, InstallShield Professional 5.x and later media for setup information that may not have been captured during repackaging, such as possible additional files, .ini files, shortcuts, or registry data (such as for language-specific setups or platform-specific setups). This primarily involves OS-dependent and language-dependent files that may be included in header files. Ultimately, this makes the installation more portable.

The SmartScan Wizard is also able to group files, shortcuts and registry entries into features corresponding to InstallShield Professional components. Any items (files, folders, shortcuts, or registry entries) that are attached to a component directly or indirectly (through File Groups), are attached to their corresponding features.

The first panel in the SmartScan Wizard is the **Welcome** panel.



Figure 8-19: SmartScan Wizard Welcome Panel

When you click **Next**, you advance to the **Original InstallShield Professional Setup Panel**, where you specify the InstallShield Professional setup you want to scan.



Note • For your convenience, if a Repackaging project is opened (either by using Open on the File menu or by using the Repackaging Wizard), and the project meets certain conditions, the SmartScan Wizard is automatically launched. The SmartScan Wizard is launched:

- if the original setup was an InstallShield Professional 5.5 or later installation, AND
- the file has not already been repackaged using the [InstallShield Professional Logging Method](#) (available for InstallShield Editor and DevStudio 9.x InstallScript installations only), AND
- the SmartScan Wizard has not already been run on this project

Original InstallShield Professional Setup Panel

On this panel, you specify the InstallShield Professional setup executable that you want to scan, and the original target directory of that executable.

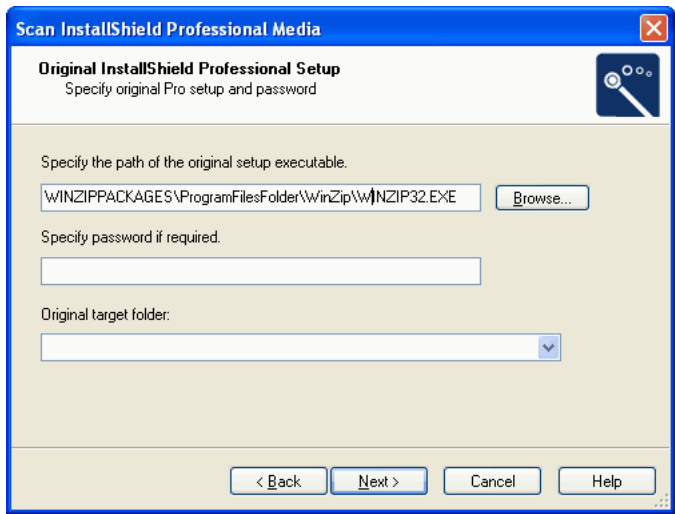


Figure 8-20: SmartScan Wizard Original InstallShield Professional Setup Panel

Enter the following options:

Table 8-24 • Original InstallShield Professional Setup Panel Options

Option	Description
Specify the path of the original setup executable	This field is filled in by default if that information exists. Click Browse to specify the InstallShield Professional setup executable that you want to scan. If you enter an invalid path name, the Next button will be disabled.
Specify password if required	Specify a password of the path of the original setup executable, if required.

Table 8-24 • Original InstallShield Professional Setup Panel Options (cont.)

Option	Description
Original target folder	<p>By default, this field is populated with information found in the original setup package. Enter the directory where the product was installed during the repackaging process. In most cases, this will be a subdirectory of [ProgramFilesFolder]. Alternatively, you can enter any target where the product was installed during the repackaging process.</p> <p>For example, if you installed this product to C:\Program Files\MyCompany\MyProduct Folder, you would enter the following in this field:</p> <p>[ProgramFilesFolder]MyCompany\MyProduct</p>

Click Next to proceed to the **Scanning Panel**, which shows the output and progress of the scan.

Scanning Panel

This panel displays the progress that SmartScan Wizard is making as it scans an InstallShield Professional setup.

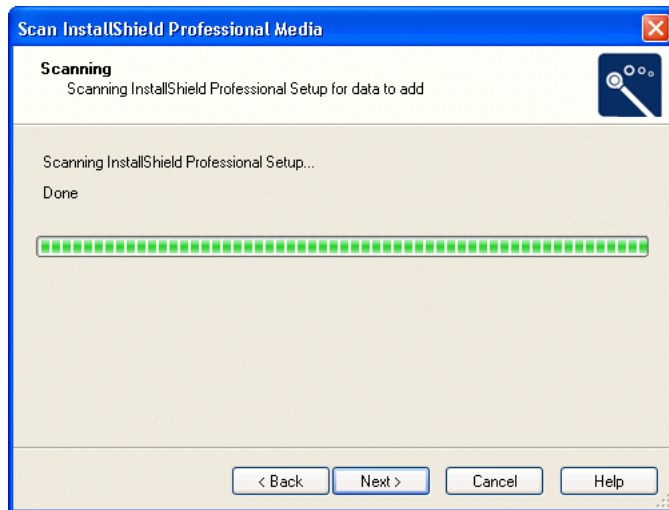


Figure 8-21: SmartScan Wizard Scanning Panel

Setup Feature Tree Panel

The Setup Feature Tree Panel lists the features and components that were configured in the original InstallShield Professional project. If you want to use this feature and component information, select the **Add feature tree** option.

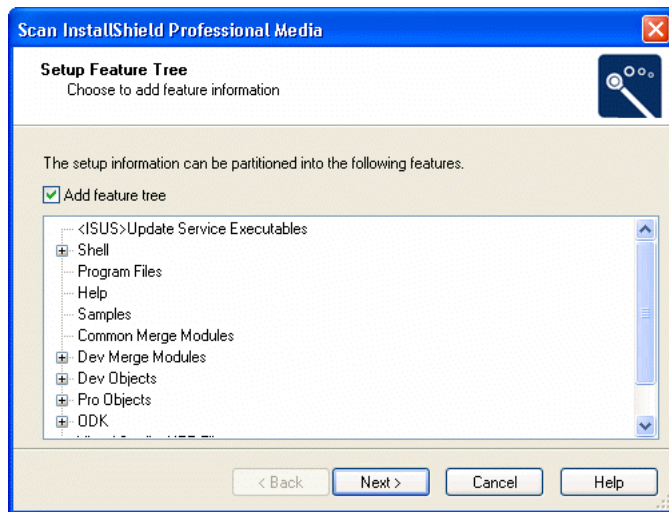


Figure 8-22: SmartScan Setup Feature Tree Panel

Click **Next** to proceed to the **Scanning Media Panel**, which first scans the original media, then displays the results.

Scanning Media Panel

Following the actual scan of the original setup media, the SmartScan Wizard displays a log of files to be included in the project.

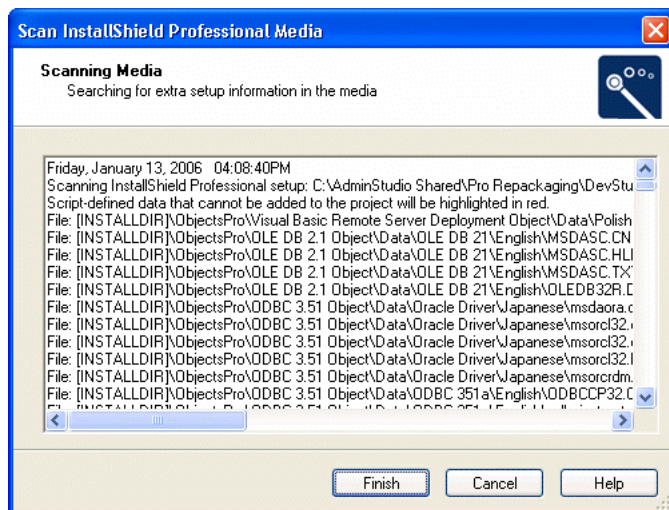


Figure 8-23: SmartScan Wizard Scanning Media Panel

Click **Finish** to add these to the project. Any default exclusions based on the exclusions file will be applied.

Setup Intent Wizard

Although an installation may have intended to install certain files, these files sometimes may not be installed—often because the files already exist on the target machine (either as the same version or a newer version). These files, although not installed or updated, are needed for the product to execute properly when the setup is run on a system that does not already have these files.

The Setup Intent Wizard allows you to scan a setup to identify files that may not have been captured during repackaging—effectively recognizing the installation's intent for these files.



Tip • Any files found will be displayed in Repackager in a different color (as specified in the **Color** tab of the Options dialog box).

The Setup Intent Wizard consists of the following panels:

- Welcome Panel
- Scanning Project Panel
- Results Panel

Welcome Panel

The first panel in the Setup Intent Wizard informs you the purpose of the Wizard, and warns you the source files for your project must be present for successful scanning.

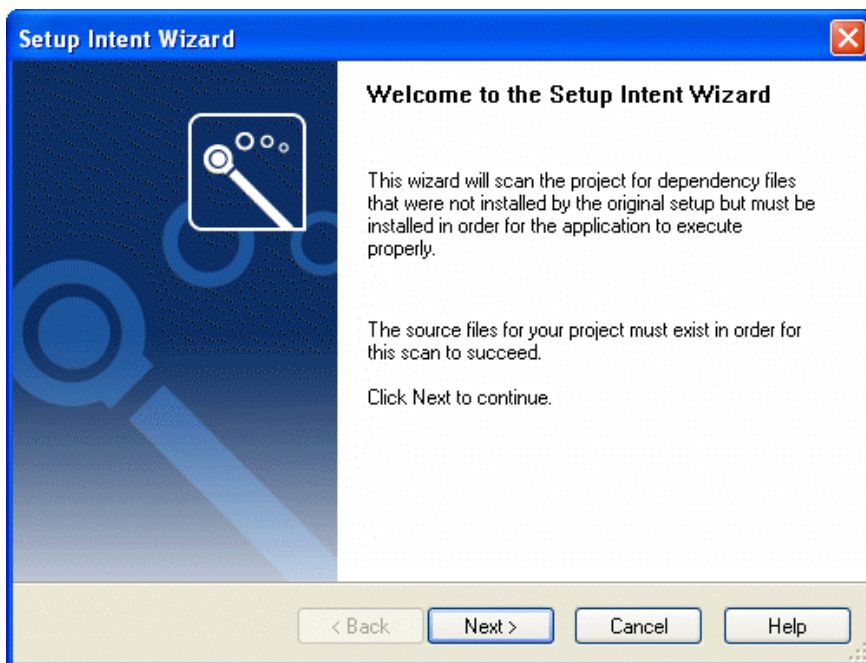


Figure 8-24: Setup Intent Wizard Welcome Panel

Click Next to start the scan and display the **Scanning Project Panel**.

Scanning Project Panel

The Scanning Project Panel is displayed while scanning is in progress. Each file scanned is listed, and a progress bar displays the overall scan progress.

When the scan is complete, the **Results Panel** opens, listing new files that your setup required.

Results Panel

The final panel in the Setup Intent Wizard allows you to view and select new files detected by the Wizard, but not already included in your Repackaging project.

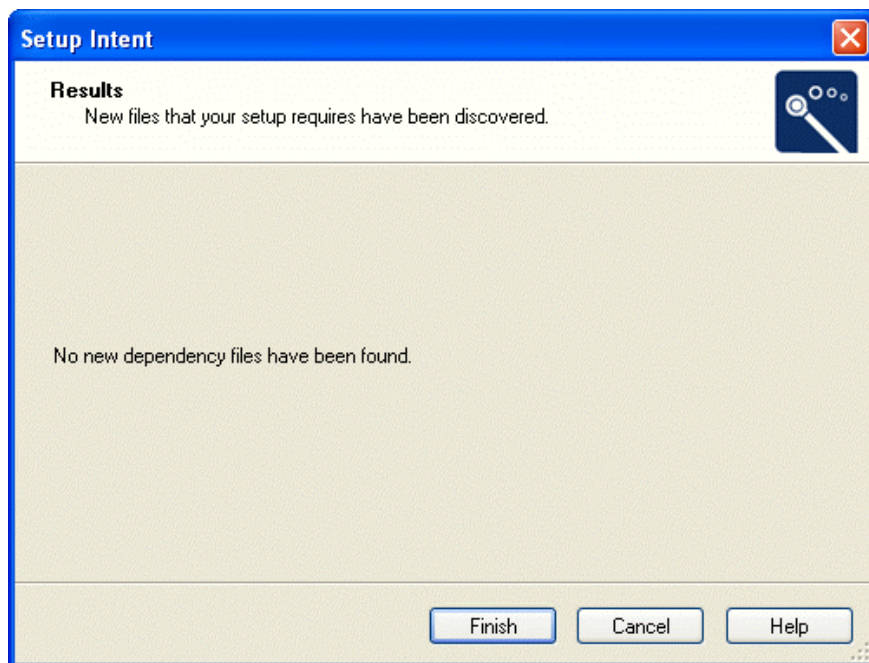


Figure 8-25: Setup Intent Wizard Results Panel

Select the files you want to include in your project which were not identified during repackaging.

Click **Finish** to exit the Setup Intent Wizard and return to the Repackaging project (with selected files automatically added to the project), or click **Back** to return to the **Scanning Project Panel**.

VMware Repackaging Wizard

Repackager includes integration with VMware Workstation's virtual machine technology. This provides you with the ability to launch a VMware session for repackaging purposes, and run different operating systems on the same computer. By using VMware, you are able to forego the traditional "ghosting" for clean images each time a new application is repackaged by simply electing not to save changes to the VMware session. You can then reload the clean state of the operating system, and proceed to the next package.



Note • AdminStudio supports VMware 3.0 and later.



Note • The **VMware Repackaging Wizard** menu item on the **Tools** menu is enabled if Repackager finds VMware 3.0 and later installed on the workstation, and if a VMware image exists on that machine. If no VMware images are found, the VMware Repackaging Wizard menu item will be disabled. Repackager reads the information about VMware images from:

`<Application Folder>\VMware\Virtual Machines.vmls`

Virtual Machines.vmls is a text file that contains information about individual VMware images and where the configuration file for each image is located. This file should contain information for at least one VMware image for the **VMware Repackaging Wizard** menu item to be enabled.

Using the VMware Repackaging Wizard, you select an available VMware operating system, and then Repackager automatically launches the selected operating system within a VMware session.

The VMware Repackaging Wizard includes two panels:

- [Welcome Panel](#)
- [VMware Virtual Machines Panel](#)

Welcome Panel

The first panel displayed in the VMware Repackaging Wizard is the Welcome panel. It explains the purpose of this Wizard: to display available VMware images on the current workstation, allowing you to select and launch the one you need.

VMware Virtual Machines Panel

On the VMware Virtual Machines panel, you select a VMware virtual machine available on the current workstation. Repackager automatically launches the selected virtual machine operating system within a VMware session so that you can begin repackaging in that environment.



Note • AdminStudio supports VMware 3.0 and later.

Click Back to return to the **Welcome Panel**; click Launch to launch the selected VMware image.

Exclusions Editor Interface

The following topics cover each tab, menu, and dialog box in the Exclusions Editor:

- [Menus](#)
- [Files Tab](#)
- [.ini Files Tab](#)
- [Registry Tab](#)
- [File Exclusion Information Dialog Box](#)
- [INI File Exclusion Information Dialog Box](#)
- [Choose Registry Key Dialog Box](#)
- [Edit Registry Key Dialog Box](#)
- [About Exclusions Editor Dialog Box](#)

Menus

Menus are not available when running the Exclusions Editor from within Repackager. They are only available when you launch the Exclusions Editor by opening the following file:

AdminStudio Installation Directory\Repackager\AnalysisOptions.exe



Note • See [Launching Exclusions Editor](#) for more information.

The following table provides a description of each menu command:

Table 8-25 • Exclusions Editor Menu Commands

Menu	Command	Keyboard Shortcut	Description
File	New	Ctrl+N	Creates a new, blank settings file.
File	Open Shared Exclusions		Opens the settings file (isrepackager.ini) from the AdminStudio Shared directory. Open this settings file when working in a team environment where the exclusion list needs to be stored in a centralized location.
File	Open Custom Exclusions		Allows you to browse to an Exclusions Editor settings file and open it. You would create a custom exclusion file based upon your company's requirements.

Table 8-25 • Exclusions Editor Menu Commands

Menu	Command	Keyboard Shortcut	Description
File	Save	Ctrl+S	Saves the current Exclusions Editor settings file.
File	Save As		Saves the current Exclusions Editor settings file to the name and location specified.
File	Exit		Exits the Exclusions Editor.
Help	Help Library		Displays the online Help Library.
Help	About Exclusions Editor		Displays the About Exclusions Editor dialog box.

Files Tab

File exclusions for Repackager indicate which files are automatically marked as excluded in the Repackager project. File exclusions in the OS Snapshot Wizard indicate files that will be excluded from the captured OS snapshot.

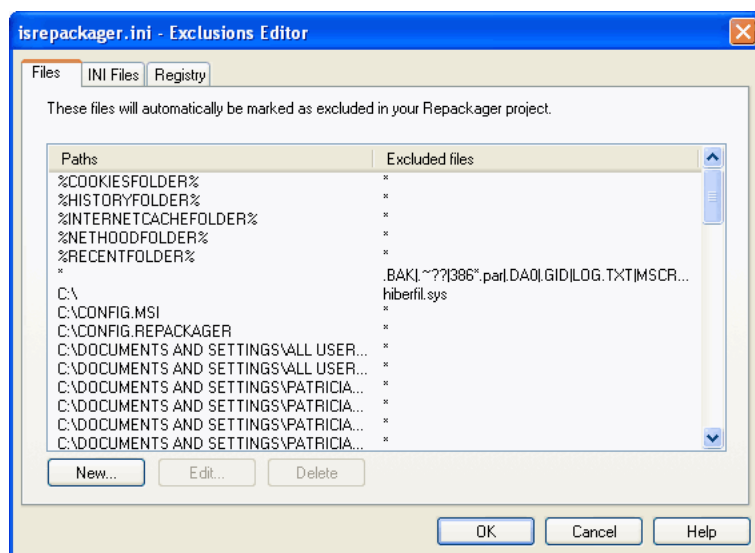


Figure 8-26: Exclusions Editor Files Tab

This **Files** tab contains a list of paths and files currently excluded from the capture process. Specific files, file extensions, and the entire contents of specified directories can be excluded.

The following three buttons allow you to add, edit, and remove files and directories from the exclusion list:

Table 8-26 • Exclusions Editor / Files Tab Buttons

Button	Description
New	Displays the File Exclusion Information dialog box from which you can specify additional file exclusions.
Edit	Brings up a dialog box from which you can change settings for the currently selected path in the exclusion list.
Delete	Deletes the currently selected path from the exclusion list.



Note • It is highly recommended that you do not edit the default exclusions for the OS Snapshot Wizard.

.ini Files Tab

.ini file exclusions for Repackager indicate which .ini files and sections are automatically marked as excluded in the Repackager project. .ini file exclusions in the OS Snapshot Wizard indicate .ini files and sections that will be excluded from the captured OS snapshot.

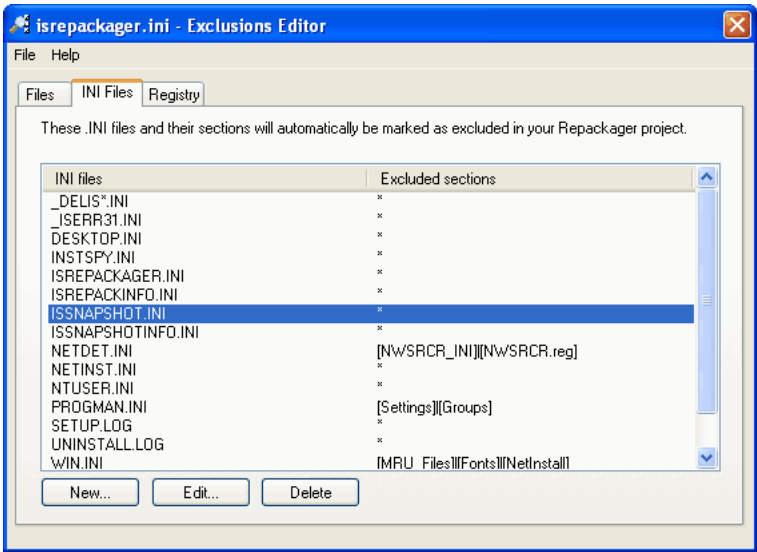


Figure 8-27: Exclusions Editor INI Files Tab

The **INI Files** tab contains a list of the .ini files and sections within .ini files excluded during analysis. If all sections are excluded, an asterisk (*) is used in the Excluded Sections column.

The following three buttons allow you to add, edit, and remove .ini files from the exclusion list:

Table 8-27 • Exclusions Editor / .ini Files Tab Buttons

Button	Description
New	Displays the INI File Exclusion Information dialog box from which you can specify additional .ini file exclusions.
Edit	Brings up a dialog box from which you can edit currently excluded .ini files.
Delete	Deletes the selected .ini file from the exclusion list.



Note • It is highly recommended that you do not edit the default exclusions for the OS Snapshot Wizard.

Registry Tab

Registry exclusions for Repackager indicate which registry keys are automatically marked as excluded in the Repackager project. Registry exclusions in the OS Snapshot Wizard indicate registry keys that will be excluded from the captured OS snapshot.

The **Registry** tab contains a list of keys and values to be excluded during registry analysis. For keys that have specific values excluded, the value name appears in the Value column. For keys that have all values excluded, an asterisk (*) represents the entire key in the Value column.

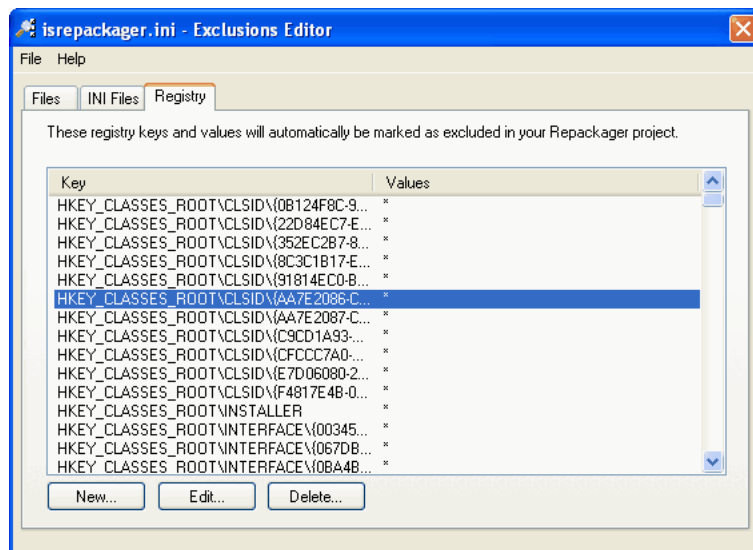


Figure 8-28: Exclusions Editor Registry Tab

There are three buttons available from this dialog box that are used to add, edit, or remove keys from the exclusion list:

Table 8-28 • Exclusions Editor / Registry Tab Buttons

Button	Description
New	Displays the Choose Registry Key dialog box, from which you can select registry keys and values for exclusion during analysis.
Edit	Brings up a dialog box from which you can modify the selected key's exclusion settings.
Delete	Removes the selected key from the exclusion list.



Note • It is highly recommended that you do not edit the default exclusions for the OS Snapshot Wizard.

File Exclusion Information Dialog Box

The **File Exclusion Information** dialog box, which is accessed by clicking **New** or **Edit** on the **Files Tab**, allows you to specify files to be excluded from analysis by the capture tool.

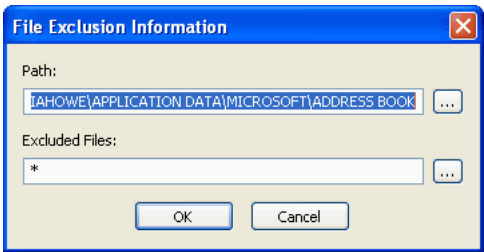


Figure 8-29: File Exclusion Information Dialog Box

Click the Browse button next to the **Path** field and select the directory that contains the file or files you want to exclude. Then, click the Browse button next to the **Excluded Files** field and select the file or files you want to exclude. In the **Excluded Files** field, you can specify files to exclude in the following ways:

- To exclude multiple files from the same directory, separate the file names with pipes (|), such as:
`file.dll|myfile.exe|anotherfile.exe`
- To exclude all files with a certain extension in the selected directory, enter an asterisk (*) plus the extension, such as:
`*.txt`
- To exclude all files in the selected directory, enter an asterisk (*) .

Click **OK** to return to the **Files Tab**.

INI File Exclusion Information Dialog Box

The INI File Exclusion Information dialog box, which is accessed by clicking **New** or **Edit** on the [.ini Files Tab](#), allows you to specify .ini files to be excluded from analysis by the capture tool.

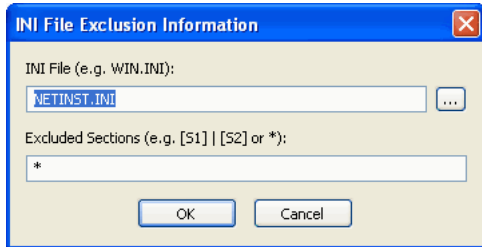


Figure 8-30: INI File Exclusion Information Dialog Box

Enter or browse to the .ini file you want to exclude, and provide the section(s) to be excluded. Sections must be enclosed in square brackets ([]), and separated by vertical bars (|) if more than one section in an .ini file is to be excluded (for example, [Groups],[Settings]). You can also exclude all .ini file sections by only entering an asterisk in the Excluded Sections field.

Click **OK** to return to the [.ini Files Tab](#).

Choose Registry Key Dialog Box

The Choose Registry Key dialog box, which is accessed by clicking **New** on the [Registry Tab](#), provides a way for you to select registry keys that you want excluded from analysis.

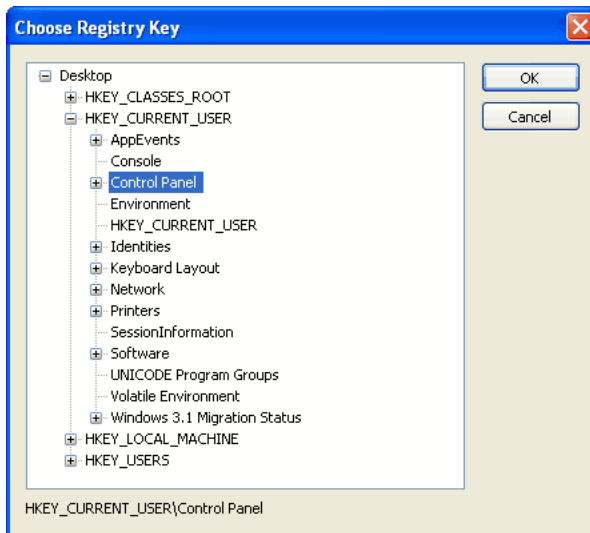


Figure 8-31: Choose Registry Key Dialog Box

Navigate through the tree until you find the key you want to exclude and click **OK** to return to the [Registry Tab](#).

By default, all values in that key are excluded. To modify this, select the key from the [Registry Tab](#) and click **Edit** to display the **Edit Registry Key** dialog box.



Tip • You can also select a registry hive to exclude. As with individual registry keys, all values (and keys) contained in the hive are excluded by default.

Edit Registry Key Dialog Box

When you select a registry key on the [Registry Tab](#) and click **Edit**, the **Edit Registry Key** dialog box opens.

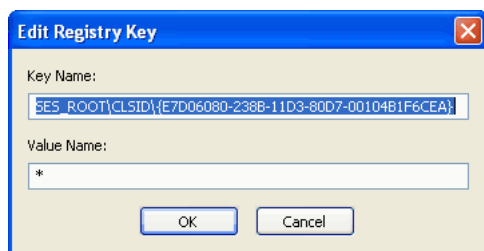


Figure 8-32: Edit Registry Key Dialog Box

You can modify the **Key Name** and/or **Value Name** excluded during analysis.

Click OK to return to the [Registry Tab](#).

About Exclusions Editor Dialog Box

The About Exclusions Editor dialog box displays version and copyright information for the Exclusions Editor. This may be useful if you need to report a problem encountered when using the Exclusions Editor.

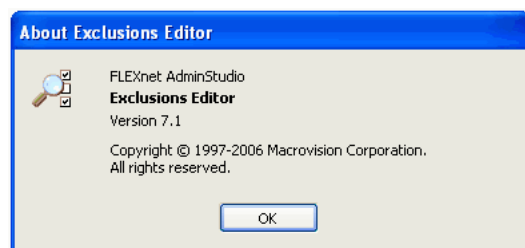


Figure 8-33: About Exclusions Editor Dialog Box

Options.ini File

The Options.ini file is created by Repackager and is used during the conversion of Repackager output into an InstallShield Editor project (.ism). It includes basic project settings which are required by Repackager. Information about this file is presented in the following sections:

- [\[MMExclusions\] Section](#)
- [\[General\] Section](#)
- [\[IgnoreShortcuts\] Section](#)
- [Options.ini File Defaults](#)



Note • Although many of these settings have a one-to-one correspondence with settings available in the Repackager interface, some can only be accessed by editing this .ini file directly.

[MMExclusions] Section

This section lists the merge module GUIDs that should not be included in your package. This section only applies if you have selected to replace files with merge modules during conversion.

[General] Section

Following are descriptions of properties that can be set in the [General] section of the Options.ini file.

Table 8-29 • Options.ini File/General Section Properties

Properties	Description
AddIMMSearchPath	Use to specify additional directories containing custom merge modules to use during repackaging.
ALLUSERS	<p>If this option is set to Y and if the template file (specified using the <code>ProjectTemplate</code> option) does not contain ALLUSERS in its Property table, then a property named ALLUSERS with a value of 2 will be added to the Property table. This will cause silent installs to behave as non-silent installs do (non-silent installs rely on a custom action to set this property).</p> <p>This option is set to Y by default.</p>
ARPPublisher	This populates the Publisher field in Add/Remove Programs in the Control Panel.
ARPPublisherURL	This populates the Publisher URL field in Add/Remove Programs in the Control Panel.
ARPSupportURL	This populates the Support URL field in Add/Remove Programs in the Control Panel.
AutoUpgrade	<p>Upgrades the InstallShield Editor template project file (if used) if needed.</p> <p>This option is set to Y by default.</p>

Table 8-29 • Options.ini File/General Section Properties (cont.)

Properties	Description
BuildCompressed	This option, set to Y by default, compresses all necessary files inside the MSI package, as opposed to storing them outside of the MSI database.
BuildFeatures	If the SmartScan Wizard is specified to include feature information in this project, this option will be set to Y . This option is set to Y by default.
BuildMSI	Specifies whether or not to build the MSI package after building ISM. This option is set to Y by default.
BuildProduct	Identifies the InstallShield Editor Product configuration to build.
BuildProScannedFiles	Files identified in the Media Scan Wizard will be included in the package (unless you have manually excluded them from the project). This option is set to Y by default.
BuildRelease	Identifies which InstallShield Editor Release configuration to build.
BuildStaticScannedFiles	Any files identified when running the Setup Intent Wizard will be included in the package (unless you have manually excluded them from the project). This option is set to Y by default.
COMMapping	When this option is set to Y , registry data pertaining to COM information will be mapped to the appropriate MSI tables whenever possible. This reduces the number of ICE33 warnings that can occur during package validation. This option is set to Y by default.
CreateSetupExe	This option, which is set to N by default, allows you to automatically create a Setup.exe file to begin the installation.
EnablePathVariables	Set this option to Y to use path variables. If enabled, the repackaged setup is significantly more portable between computers (with dependencies to the system where the setup was repackaged removed). This option is set to Y by default.

Table 8-29 • Options.ini File/General Section Properties (cont.)

Properties	Description
ExtraHKCRPermanent	<p>When this option is set to Y, any changes made to existing registry data during repackaging which cannot be identified as belonging to a file installed by the setup are placed in permanent components, which are not removed by default when the repackaged setup is uninstalled. This prevents inadvertently removing registry entries required by other applications that were not originally made by the repackaged setup.</p> <p>By default, this option is set to Y, and it is strongly recommended that you retain this setting to prevent unexpected results when the package is uninstalled.</p>
INSTALLDIR	<p>This value will be used for INSTALLDIR (the installation directory) and can use a Windows Installer property such as</p> <p>[ProgramFilesFolder]\MyProgram</p>
ISProSetup	<p>If one of the original setups that was repackaged was created by InstallShield Professional 5.5 or later, this option will be set to Y.</p> <p>This option is set to N by default.</p>
LimitedUI	<p>Set this option to Y display only the InstallWelcome dialog box when the MSI package is run.</p> <p>This option is set to Y by default.</p>
MultiUserShortcuts	<p>When this option is set to Y, non-advertised shortcuts will work for all users on the target system. This will generate ICE43 warnings when validation is run. If you know the installation is for a single-user environment, change this option to N to avoid these warnings.</p> <p>This option is set to Y by default.</p>
MMPathVersion	<p>When including merge modules, if this option is set to Y, compare path and version information.</p> <p>This option is set to Y by default.</p>
NewInstallDir	Value for INSTALLDIR variable.

Table 8-29 • Options.ini File/General Section Properties (cont.)

Properties	Description
ODBCMapping	<p>If selected, ODBC-related registry data is mapped to ODBC tables instead of the Registry table. This data will only function correctly if Windows Installer supports the ODBC resource being mapped; it is recommended that you do not enable this option if you are unsure whether the ODBC resources are supported correctly by Windows Installer.</p> <p>This option is set to N by default.</p>
OSGranular	<p>For SmartScanned or ProLogged Projects:</p> <ul style="list-style-type: none"> • If this is set to Y, component conditions will store specific operating system information. For example, if the filter is NT4, the condition will be (VersionNT=4). • If this is set to N, component conditions will store a grouping of the operating system. For example, if the filter is NT4, the condition will be (VersionNT). <p>This is set to N by default.</p>
OtherComponentFileExtensions	<p>Specify additional extensions to use when defining components. MSI has rules governing component creation for file types. For example, portable executable (PE) files must have separate components. Therefore, certain extensions have been defined (EXE, DLL, etc.). Additional extensions can be defined in the options.ini file in the format of:</p> <p>Type1:Extension1 Type2:Extension2</p> <p>where Type is one of the following numbers:</p> <p>0 = other 1 = PE 2 = help 3 = font 4 = INI</p> <p>This option is set to 1:QTX 1:AX by default.</p>
OtherFilesNewComponents	<p>When this option is set to Y, one component will be created for every file in your setup. Otherwise, new components will only be created for each portable executable file.</p> <p>This option is set to N by default.</p>

Table 8-29 • Options.ini File/General Section Properties (cont.)

Properties	Description
PermanentSystemFiles	Set this option to Y to mark portable executable files installed to a system folder (System32Folder) as Permanent files (will not be uninstalled). This option is set to Y by default.
PermanentSystemFilesSubfolders	Set this option to Y to mark files installed to a subfolder of a system folder as Permanent files (will not be uninstalled). This option is set to N by default.
ProductName	The name of the product. You must provide a value for this option either in this file or in Repackager.
ProductVersion	The version of the product. You must provide a value for this option either in this file or in Repackager.
Project	Name of InstallShield Editor project file.
ProjectTemplate	The name and location of the default InstallShield Editor project template (.ism) used in the conversion process.
ServiceControlEvents	When this option is set to Y, the ServiceControl table will be populated for NT Services. This option is set to N by default.
SharedCommonFiles	Set this option to Y to mark portable executable files installed to the CommonFilesFolder (or subfolder) as Shared files. This option is set to Y by default.
SISAuthor	This option populates the Author field of the Summary Information Stream (accessible from the package's properties). This option is set to Repackager by default.
SISSubject	This option populates the Subject field of the Summary Information Stream (accessible from the package's properties).
SkipMMIfShortcut	Merge Modules that have files pointed to by shortcuts should be skipped even if they are not in the exclusion list. This option is set to Y by default.

Table 8-29 • Options.ini File/General Section Properties (cont.)

Properties	Description
SmartScannedOnce	Specifies whether or not the SmartScan Wizard was run for this project. This option is set to N by default.
UseAdvertisedShortcuts	Create advertised shortcuts where applicable. This option is set to Y by default.
UseHKCUProxy	Set this option to Y to copy all registry entries in HKEY_CURRENT_USER to HKEY_USERS\default. This option is set to N by default.
UseLanguage	When selected, the target package's language will be the language detected by Repackager (as displayed in the Captured Installation view). This option is set to N by default.
UseMergeModules	Set this option to Y to replace files with merge modules whenever possible during conversion. Exceptions are listed under the [MMExclusions] section. This option is set to Y by default.
UseSrcFolder	Set this option to Y to make the InstallShield Editor project (.ism) folder default to the Repackager output project (.inc) folder. This option is set to Y by default.

[IgnoreShortcuts] Section

Shortcuts that refer to executables listed in this section will be ignored during conversion.

Options.ini File Defaults

This section lists the default settings in the Options.ini file that is shipped with Repackager:

```
[MMExclusions]

[General]
UseSrcFolder=Y
EnablePathVariables=Y
UseHKCUProxy=N
LimitedUI=Y
SISAuthor=InstallShield Repackager
OtherFilesNewComponents=N
UseMergeModules=Y
SharedCommonFiles=Y
```

```

PermanentSystemFiles=Y
PermanentSystemFilesSubfolders=N
ExtraHKCRPermanent=Y
COMMapping=Y
ODBCMapping=N
ServiceControlEvents=N
ALLUSERS=Y
ProjectTemplate=
BuildCompressed=Y
CreateSetupExe=N
MultiUserShortcuts=Y
SmartScannedOnce=N
ISProSetup=N
BuildFeatures=Y
OtherComponentFileExtensions=1:QTX|1:AX
OSGranular=N
MMPathVersion=N

[IgnoreShortcuts]
TargetExe1=isuninst.exe
TargetExe2=uninst.exe
TargetExe3=setup.exe
TargetExe4=uninst.dll
TargetExe5=rnuninst.exe

```

Files Associated with Repackager

Several files are associated with Repackager. Some are output files, and some contain default information for Repackager to function. These files are described in the tables below.

Files Used By the Repackaging Wizard

The following files are used by the Repackaging Wizard.

Table 8-30 • Files Used by the Repackaging Wizard

File	Location	Description
Repack.ini	<p><i>Windows</i></p> <p>If the file is not found in the Windows directory, then the Repackaging Wizard extracts a default file from the resource and stores it in the Windows directory.</p>	<p>This is an input file for the Repackaging Wizard. It contains a list of the exclusions for the files, folders, .ini files and registry entries for the last used configuration of Repackager. During the Snapshot and Install Monitoring modes of repackaging, the entries in this file are filtered out from the repackaged output.</p> <p>See Repack.ini File for more information.</p>

Table 8-30 • Files Used by the Repackaging Wizard (cont.)

File	Location	Description
Options.ini	<p>Repackager output directory (specified in the Set Target Project Information and Capture Settings Panel).</p> <p>The Repackaging Wizard makes a copy of the default options.ini that is present in the following directory:</p> <p><i>AdminStudio Installation Directory\Repackager</i></p> <p>and saves this file in the same location as the current repackaged output file (.inc). Additionally, the UseSrcFolder flag can be used to store the created InstallShield Editor file in the same directory as the .inc file.</p>	<p>This is an output file from the Repackaging Wizard. It contains configuration information about the repackaged setup, including whether to use path variables, whether to display a limited user interface during installation of the repackaged setup and whether every file will go into its own component.</p>
productname.inc	<p>Created in the Repackager output directory (specified in the Set Target Project Information and Capture Settings Panel).</p>	<p>This is an output file from the Repackaging Wizard. It contains the locations of files, .ini files, and shortcuts detected by Repackager as having been created, modified, or removed during repackaging. Also, it contains a link to the standard.nir and deleted.isr files for registry information.</p>
updated.isr	<p>Created in the Repackager output directory (specified in the Set Target Project Information and Capture Settings Panel).</p>	<p>This is an output file from the Repackaging Wizard when the Install Monitoring method is used. It contains registry additions and modifications detected during repackaging using installation monitoring only.</p>
deleted.isr	<p>Created in the Repackager output directory (specified in the Set Target Project Information and Capture Settings Panel).</p>	<p>This is an output file from the Repackaging Wizard. It contains registry deletions detected during repackaging using Installation Monitoring and Snapshot.</p>
standard.nir	<p>Created in the Repackager output directory (specified in the Set Target Project Information and Capture Settings Panel).</p>	<p>This is an output file from the Repackaging Wizard when the Snapshot method is used. It contains registry additions and modifications detected during repackaging using the Snapshot method.</p>

Table 8-30 • Files Used by the Repackaging Wizard (cont.)

File	Location	Description
*.spy	Created in the following folder: <i>WindowsDrive\InstallHook</i>	This is an output file from the Repackaging Wizard when the Install Monitoring method is used. It contains API call logs for installation monitoring.
Default.ini	<i>AdminStudio Installation Directory\Repackager</i>	Contains the default configuration for Repackager, including default exclusion information.
Repack.log	<i>WindowsFolder</i>	Log file created by the Repackaging Wizard.

Files Used By the Repackager Interface

Table 8-31 • Files Used by the Repackager Interface

File	Location	Description
*.irp	Saved in the same location as the .inc file.	This is a Repackager project file. It is the main file for each repackaged or converted setup. It contains information about the .inc files referred to and also stores the file, folder, .ini files and registry exclusions made in the Repackager Interface.
<Exclusion List>.ini	varies	This is an input file for the conversion of the .inc file to an MSI package. It contains the list of files, folder, .ini files and registry entries exclusions. Users can choose a different exclusion file from the Repackager Interface and the exclusions will be reflected in the Interface.
Options.ini	Saved in the same location as the .inc file.	This is an input file for the conversion of the .inc file to an MSI package. It contains configuration information about the repackaged setup, including whether to use path variables, whether to display a limited user interface during installation of the repackaged setup, and whether every file will go into its own component. Additionally, the UseSrcFolder flag can be used to store the created InstallShield Editor file in the same directory as the .inc file.

Repack.ini File

The Repack.ini file is the default capture exclusion file for the Repackaging Wizard. It contains exclusions to be applied during repackaging, and mainly focuses on specific items that should not be included in applications, such as InstallShield Professional-specific COM settings, OS settings, and Internet Explorer settings. Any item excluded during capture will not be available for exclusion/inclusion in the Repackager project file.

The file is located in the Windows folder, and can be edited using the Exclusions Editor, or using a text editor.



Note • It is strongly recommended that you not modify this file, as it increases the likelihood of either inadvertently omitting necessary pieces of applications you are repackaging, or including registry entries or files that should not be part of the repackaged application. In the first scenario, you may need to recapture your application; in the second, you may need to exclude more from the Repackager project.

Instead, capture your application using the default exclusions in the Repackaging Wizard, and then selectively exclude captured data using the Repackager Interface. This way, if you inadvertently exclude a necessary piece, you need only reinclude it in Repackager—not recapture the application entirely.

Using InstallShield to Chain Multiple Windows Installer Packages Together

If your application includes more than one Windows Installer (*.msi) package, you can use InstallShield Editor to chain them together using a nested MSI Custom Action. This enables you to run multiple MSI files within a single setup process.

To do this, you open the InstallShield Editor **Custom Actions View** and use the **Custom Action Wizard**.



Task: *To add a Nested MSI Custom Action:*

1. Launch **InstallShield Editor**.
2. Open your Windows Installer package in Direct Edit Mode.
3. In the Installation Designer, expand the **Behavior and Logic** tree and select the **Custom Actions** node. The Custom Actions View opens.
4. In the middle pane, right-click **Custom Actions** and then click **Custom Action Wizard**.
5. Follow the **Nested Installations** procedure in the InstallShield Editor user documentation to create a nested MSI Custom Action.

Troubleshooting

Repackager Troubleshooting information is presented in the following topics:

- [Troubleshooting Guidelines for WinINSTALL Conversion](#)
- [Troubleshooting Guidelines for SMS Conversion](#)
- [Resolving an “Error Building Table File” Error](#)

Troubleshooting Guidelines for WinINSTALL Conversion

Use the following troubleshooting guidelines to identify and fix WinINSTALL conversion problems.

- **Repackager tool supports 6.0, 6.5, and 7.x project formats only.** For all other formats, please use the WinINSTALL LE tool available as a free download in Windows 2000 to convert to 7.x files.
- **Repackager tool cannot convert WinINSTALL .NAI files**—It can only convert WinINSTALL projects that have been converted to text (.txt).

- **All files must be available**—All the files that were available to the original WinINSTALL installation project must be available to the converted installation at the exact same locations.
- **Not all elements of a WinINSTALL installation are converted**—Because WinINSTALL installations are based on a different technology than Windows Installer, not all elements of a WinINSTALL installation are converted. Only the installation of files, registry changes, and other system changes are converted.
- **Custom logic is not converted**—Custom logic written in WinINSTALL's custom scripting language is not converted.
- **WinINSTALL environment variable assignments are not converted**—To re-add environment variable assignments in a Windows Installer installation, open the converted project in InstallShield Editor and use the Environment Variable view.
- **WinINSTALL variables are converted to a Windows Installer variable**—If the target path of a file contains a WinINSTALL variable, then the WinINSTALL variable is converted to a Windows Installer variable.
- **Specify @ variables at conversion time**—If the source path of a file in WinINSTALL contains either the @Server or @Wininstall variable, you can specify the values of these two variables at conversion time in the Repackager.
- **The WinINSTALL Preinstall and Postinstall scripts are not converted.**

Troubleshooting Guidelines for SMS Conversion

Use the following troubleshooting guidelines to identify and fix SMS conversion problems.

- **All files must be available**—All the files that were available to the original SMS installation project must be available to the converted installation at the exact same locations.
- **Not all elements of an SMS installation are converted**—Because SMS installations are based on a different technology than Windows Installer, not all elements of a SMS installation are converted. Only the installation of files, registry changes, .ini Files, ODBC, NT Services, Fonts, Shortcuts, Variables, and other system changes are converted.
- **Custom logic is not converted**—Custom logic written in SMS's custom scripting language is not converted.
- **SMS environment variable assignments are not converted**—To re-add environment variable assignments in a Windows Installer installation, open the converted project in InstallShield Editor and use the Environment Variable view.

Resolving an “Error Building Table File” Error

When building with Repackager, if you have received the following error message during the build:

ISDEV: fatal Error 5023: Error building table File

your first step is to go to the Repackager Interface and check whether the number of files installed by this setup is greater than 32,767. If it is, this error occurs because Windows Installer supports 32,767 files in the File table but the package being built exceeds this limit. See [Authoring a Large Package](#) in Windows Installer Help for more information.

If you want to fix this error using Repackager, perform the steps listed below.



Task: *To fix this error using Repackager:*

1. Browse to the appropriate directory:
 - If you are using the standalone Repackager, browse to the Repackager folder.
 - If you are using the Repackager on a machine where AdminStudio is fully installed, browse to the following directory:
`<AdminStudio INSTALLDIR>\Editor\Support\0409`
2. Locate the IsMsiPKg.itp and IsMsiPKgLarge.itp files in this directory.
3. Rename IsMsiPKgLarge.itp to IsMsiPKgLarge.itp.bak.
4. Make a copy of IsMsiPKg.itp and rename the copy IsMsiPKgLarge.itp.
5. Perform the conversion and create the MSI.
6. Delete IsMsiPKg.itp.
7. Rename IsMsiPKgLarge.itp.bak back to IsMsiPKgLarge.itp, thereby restoring the original file.



Note • *Transforms and patches cannot be created between two packages with different column types.*



Note • *For more information, see the [Authoring a Large Package](#) and [File Table](#) topics in the Windows Installer Help.*

Converting Windows Installer Packages to Virtual Applications



Important • AdminStudio ThinApp support requires a separate purchase of VMware® ThinApp™.

You can use the Repackager Windows Installer Converter to convert a Windows Installer package into the following virtual application formats: Microsoft App-V, VMware ThinApp, and Citrix XenApp.

This section presents an overview of virtualization and then explains how to create virtual applications:

Table 9-1 • Using the Windows Installer Converter

Section	Description
Getting Started With Application Virtualization	Identifies the options you have when creating a virtual application, and suggests which option you should choose—depending upon your source files and the degree of customization you want to perform.
About Microsoft Application Virtualization	Introduces you to the Microsoft App-V virtualization technology and describes the components of an App-V application.
About ThinApp Virtual Packages	Introduces you to the VMware ThinApp virtualization technology and describes the components of a ThinApp application.
About Citrix Virtual Packages	Introduces you to the Citrix XenApp virtualization technology and describes the components of a Citrix profile.
Converting a Single Application to a Virtual Package	Explains how to use the Windows Installer Converter to convert a single Windows Installer package into a virtual package.

Table 9-1 • Using the Windows Installer Converter (cont.)

Section	Description
Converting a Suite of Applications into a Virtual Package	Explains how to use the Windows Installer Converter to convert multiple Windows Installer packages—such as a group of applications that need to work together—into one virtual package.
Converting a Directory of Packages into Individual Virtual Packages	Explains how to use the Windows Installer Converter to convert a directory of Windows Installer packages into individual virtual packages.
Testing Windows Installer Packages for Virtualization Compatibility	Explains how to use the Pre-Conversion Evaluation feature of Windows Installer Converter to test applications for virtualization compatibility before proceeding with conversion.
Testing App-V Applications Using the App-V Application Launcher	Describes using the App-V Application Launcher testing tool that enables you to reliably and accurately test your App-V applications on your local machine or any other system that has the App-V client installed before moving it to the App-V server.
Running the Windows Installer Converter from the Command Line	Explains how to run Windows Installer Converter from the command line.
Capturing Virtualization Context	Explains the context file that is created during virtual application build.
Windows Installer Converter Reference	Describes each of the panels in the Windows Installer Converter, and contains a list of the error messages that could occur when building a virtual package.



Note • For information on other Repackager features, see [Repackaging Legacy Installations Using the Repackaging Wizard](#) and [Converting Legacy Installations Using the Repackager Interface](#).

Getting Started With Application Virtualization

AdminStudio provides support for the conversion of Windows Installer packages to the following virtual application formats:

- Microsoft App-V applications
- VMware ThinApp applications
- Citrix profiles, which can be deployed on Citrix XenApp

You have several options when deciding how you want to create a virtual application, depending upon your source files and the degree of customization you want to perform:

Table 9-2 • Application Virtualization Support in AdminStudio




If you have ...	And want to ...	Use ...	Description
<p>Windows Installer Package</p> 	<p>Convert it to a virtual application with ...</p> <ul style="list-style-type: none"> • No modifications • Default isolation options 	<p>Windows Installer Converter</p>	<p>Use the Windows Installer Converter to convert a Windows Installer .msi file to a virtual application.</p> <p>For detailed information, see:</p> <ul style="list-style-type: none"> • Converting a Single Application to a Virtual Package • Converting a Suite of Applications into a Virtual Package • Converting a Directory of Packages into Individual Virtual Packages • Testing Windows Installer Packages for Virtualization Compatibility
<p>Windows Installer Package or InstallShield Project</p> 	<p>Convert it to a virtual application with ...</p> <ul style="list-style-type: none"> • Modified package contents, registry settings, and shortcuts • Custom isolation options on folders and registry entries • Operating system and/or language requirements 	<p>InstallShield Editor Microsoft App-V Assistant or InstallShield Editor Citrix Assistant or InstallShield Editor ThinApp Assistant</p>	<p>Use the InstallShield Editor Microsoft App-V Assistant, ThinApp Assistant or Citrix Assistant to create a virtual application from an InstallShield project or a Windows Installer package.</p> <p>Customization options include:</p> <ul style="list-style-type: none"> • Modifying package contents, registry settings, and shortcuts • Setting custom isolation options on folders and registry entries • Setting operating system and/or language requirements • Specifying deployment server <p>For detailed information on how to use the Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant, see <i>Creating Customized Virtual Applications</i> in the InstallShield Help Library.</p>

Table 9-2 • Application Virtualization Support in AdminStudio (cont.)

If you have ...	And want to ...	Use ...	Description
Repackager Project 	Convert it to a virtual application with ... <ul style="list-style-type: none"> • No modifications • Default isolation options 	Repackager	<p>By selecting an option on the Repackaged Output view, you can simultaneously build an InstallShield Editor project, a Windows Installer package, an App-V application, a ThinApp application, and a Citrix profile from a Repackager project.</p> <p>For information on this feature, see Automatically Generating a Virtual Application During Repackager Project Build.</p>

About Application Virtualization



Note • This section provides a description of virtualization in general for those that are not familiar with it. It does not represent the architecture of any specific vendor.

A typical Windows application has dependencies on components that are shared by multiple applications, such as registry entries or COM controls. When an installation author recognizes that their application requires a shared component—such as MDAC (Microsoft Data Access Components)—they include a merge module to install that component.

When one of these shared components is installed during an application's installation, it is possible that a previously-installed version of the same component could be overwritten, causing the existing application to break. Because of these possible problems, extensive compatibility testing needs to be performed before an application can be distributed in the enterprise environment. The following diagram provides an example of two conflicting installed applications.

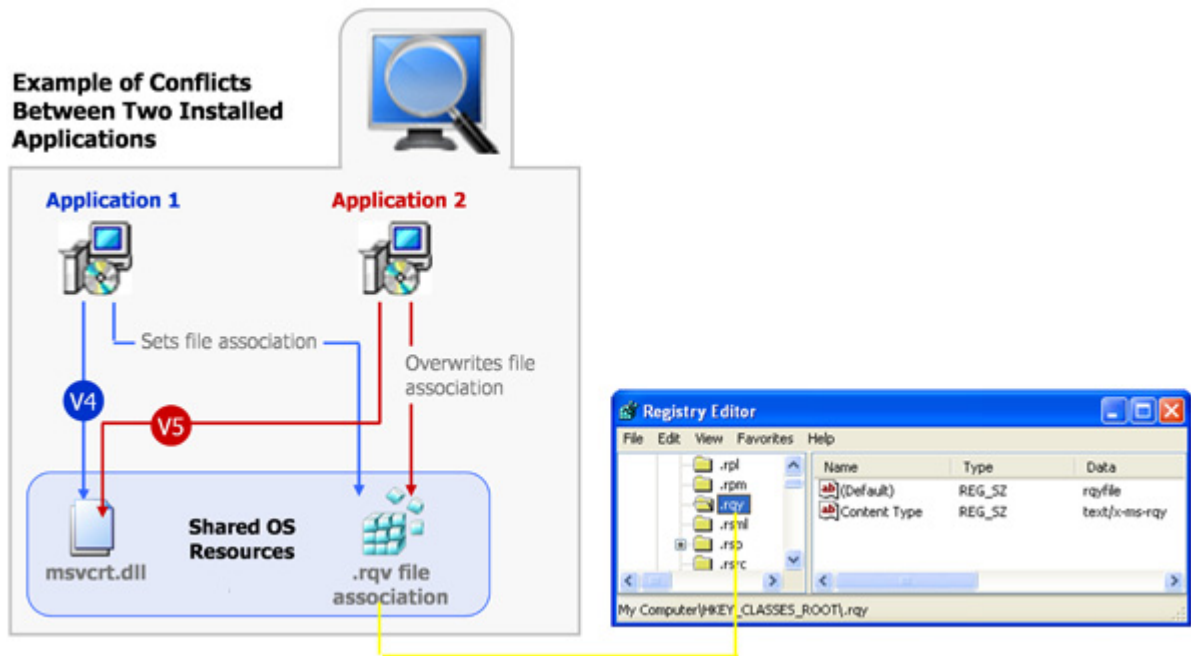


Figure 9-1: Example of Conflicts Between Two Installed Applications

Virtualization simplifies the situation by keeping the application layer and the operating system layer separate, so that the virtual application has no impact on the other applications. In application virtualization, a container or isolation environment is created around the application: a controlled virtual space for application execution that separates the interaction between an application and the underlying operating system's resources in order to protect applications from conflicting with each other.

The following diagram provides an example of how application virtualization would solve the conflicts shown in the previous example.

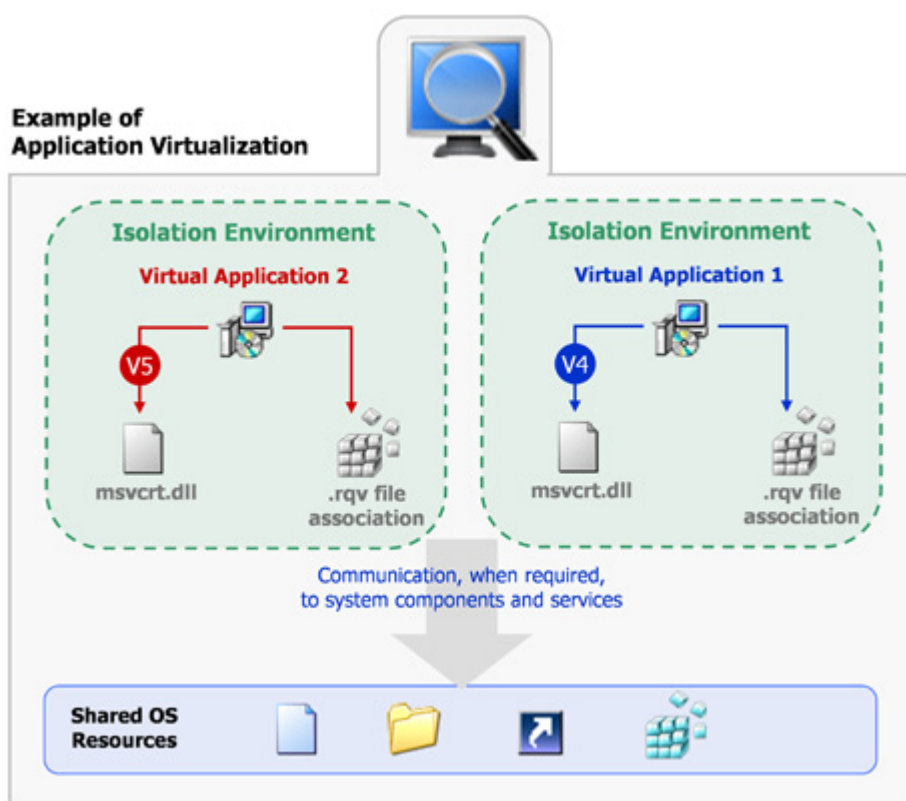


Figure 9-2: Example of Application Virtualization

Application virtualization allows the configuration of an application to be standardized to an isolation environment, rather than to an individual user's desktop machine. Application objects, files and registry settings are contained within this isolation environment. Critical application resources are managed locally by the isolation environment, thus minimizing resource dependencies between applications.

Application virtualization greatly reduces the scope for conflicts between applications and, therefore, simplifies compatibility testing.

Types of Virtual Package Conversion

When performing application virtualization, Windows Installer Converter supports several modes of conversion. You have three options when performing virtual package conversion with Windows Installer Converter:

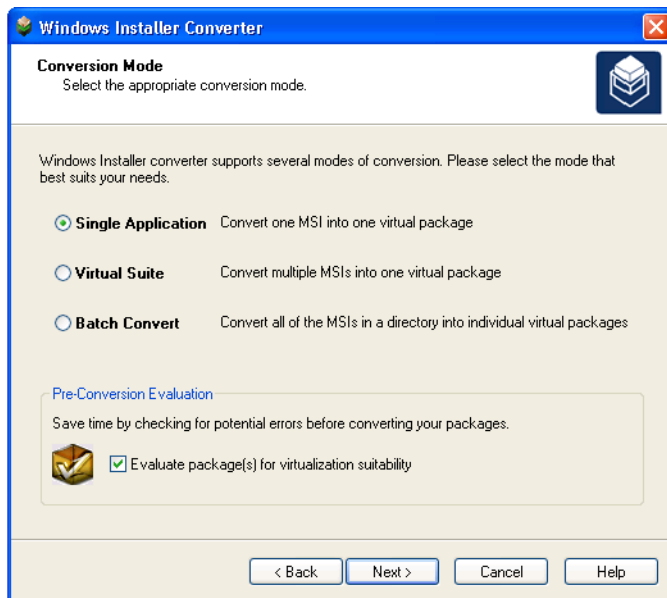


Figure 9-3: Windows Installer Converter / Conversion Mode Panel

Conversion mode options are:

- [Single Application](#)
- [Virtual Suite](#)
- [Batch Conversion](#)

In addition to using the Windows Installer Converter, you can also convert packages using Repackager, InstallShield Editor, and via command line. See [Other Methods of Conversion](#).

Single Application

Select this option if you want to convert one Windows Installer package into a virtual package.

Virtual Suite

Select this option if you want to convert multiple Windows Installer packages, perhaps a group of applications that need to work together, into one virtual package.

- **Microsoft App-V**—When creating a Microsoft App-V application, Windows Installer Converter converts the installation data from the original Windows Installer package into an .sft file that is compatible with Microsoft App-V version 4.5. The installation shortcuts are converted into .osd files, which provide access to programs within the .sft file.
- **ThinApp**—When creating a virtual suite of packages using ThinApp, Windows Installer Converter creates a program shortcut for every application in the suite and all shortcuts are presented to end users.
- **Citrix**—When creating a virtual suite of packages using Citrix XenApp, Windows Installer Converter also creates a program shortcut for every application in the suite. The .profile file contains the shortcut information, and the .cab file contains all of the files necessary to run the applications in the suite. However,

when you distribute a virtual package using Citrix XenApp, you can only select one shortcut. If you want this virtual suite to have more than one shortcut, you will need to distribute it more than once.

Batch Conversion



Edition • The **Batch Convert** option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.

You can use the **Batch Convert** option to convert a directory full of Windows Installer packages into individual virtual packages. You may already have your enterprise packages organized into a directory structure. The **Batch Convert** option allows you to simply take that entire directory structure, optionally run a test conversion, and then convert all of the applications in that directory in one operation. This operation will not modify the existing installation data files. It will perform the conversion operation and save the virtual packages in new subdirectories.



Tip • It is recommended that each of these Windows Installer packages be in a separate subdirectory of the selected directory.

When using the **Batch Convert** option, any Windows Installer transform files that are colocated in the same directory with a Windows Installer package file are identified and examined to determine if they can be applied to that Windows Installer package. Those that are confirmed are ordered by file last modified date (oldest to youngest) and are included in the virtual package.



Note • If you select the **Evaluate package(s) for virtualization suitability** option on the **Conversion Mode** panel, the package(s) is tested prior to conversion. After viewing the test results, you can choose the packages you want to convert. See [Testing Windows Installer Packages for Virtualization Compatibility](#).

Other Methods of Conversion

In addition to using the Windows Installer Converter, you can use several other methods to create a virtual application:

- **InstallShield Editor Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant**—Using one of these Assistants, you can configure file, folder, and registry key isolation options before building a virtual package.

For detailed information on how to use the Microsoft App-V, Thin-App, and Citrix Assistants, see [Creating Customized Virtual Applications](#).

- **Repackager**—You can simultaneously build an InstallShield Editor project, a Windows Installer package, and a virtual application from your Repackager project. See [Automatically Generating a Virtual Application During Repackager Project Build](#).
- **Command Line**—You can run the Windows Installer Converter from the command line. See [Running the Windows Installer Converter from the Command Line](#).

About Microsoft Application Virtualization

Microsoft Application Virtualization (App-V) provides the capability to make applications available to end user computers without having to install the applications directly on those computers. Information on Microsoft Application Virtualization is presented in the following topics:

- [About Microsoft Application Virtualization \(App-V\)](#)
- [Creating 64-Bit App-V Packages](#)
- [Components of an App-V Application](#)

About Microsoft Application Virtualization (App-V)

Microsoft Application Virtualization (App-V) enables applications to run as network services, removing the need for local installation of the applications. An App-V application runs in a self-contained, virtual environment. The virtual environment contains the information necessary to run the application on the client without installing the application locally. Only the App-V client needs to be installed on the client machines. Even though these virtual applications are never installed, they can communicate with the local operating system, middle ware, plug-ins, and other applications.

Because App-V applications are not installed on the client, there is minimal impact on the host operating system or other applications. As a result, application conflicts and the need for regression testing are dramatically reduced.

Using Microsoft Application Virtualization enables you to centralize the installation and management of deployed applications, and control access to applications. The App-V client presents to the end user a list of applications to which that user has access.

Microsoft Application Virtualization Infrastructure

The Microsoft Application Virtualization (App-V) infrastructure includes:

- **App-V Sequencer**—The App-V Sequencer converts application data into a format which is compatible with the App-V server and client, producing an App-V application.
- **App-V Server**—An App-V application can be placed on one or more App-V servers so that it can be streamed down to the clients on demand and cached locally.
- **App-V Client**—The App-V Client is the system component that enables the end user to interact with the App-V applications that are available on the App-V server.

Benefits of Using the Windows Installer Converter

Instead of using the App-V Sequencer to create App-V applications, you can use the Windows Installer Converter, as shown in the following diagram:

Chapter 9: Converting Windows Installer Packages to Virtual Applications

About Microsoft Application Virtualization

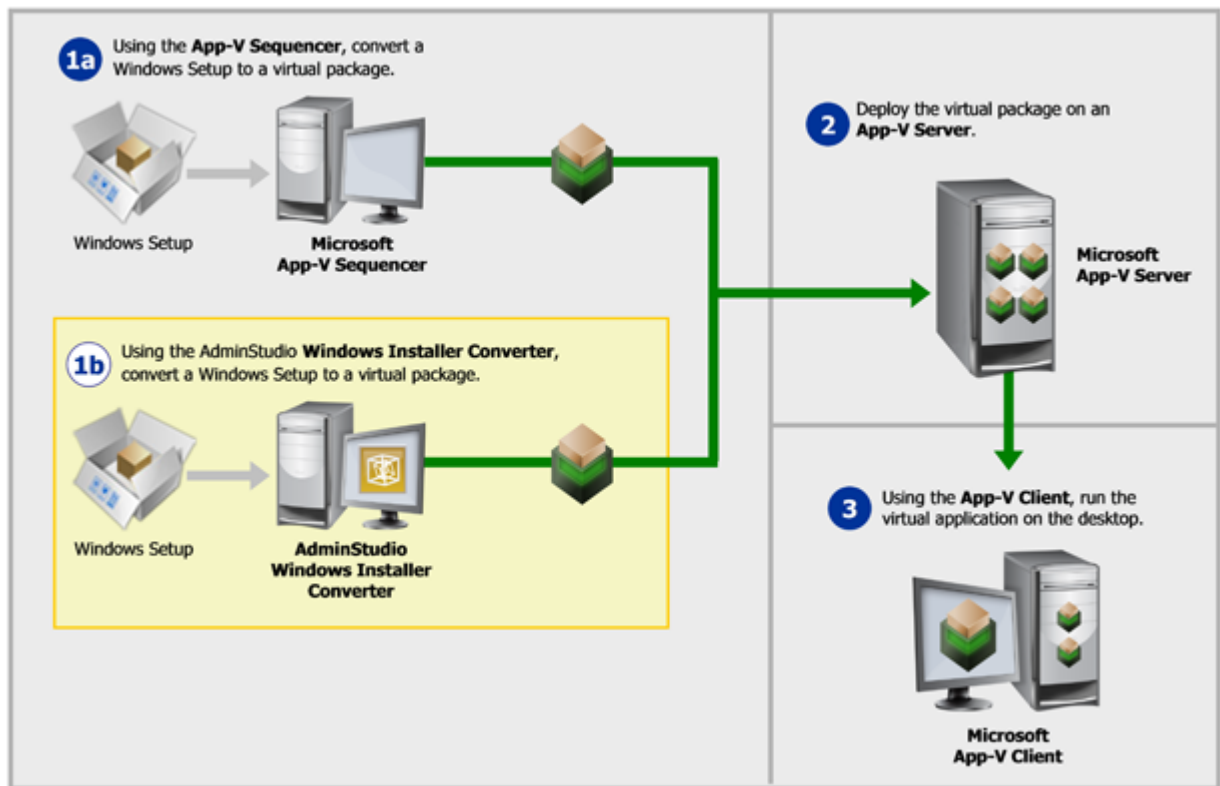



Figure 9-4: Using the Windows Installer Converter to Create an App-V Package

Using the Windows Installer Converter instead of the Microsoft App-V Sequencer to create an App-V application offers the following benefits:

Table 9-3 • Benefits of Using the Windows Installer Converter to Create an App-V Application

Windows Installer Converter Benefit	Problem	Solution
Product Installation on a Clean Machine Is Not Required	<p>The Microsoft App-V Sequencer obtains the information it needs to create an App-V application by installing a package on a clean machine and then comparing the file system snapshot that it took prior to installation with one it takes after installation. To perform this task properly, there are two requirements:</p> <ul style="list-style-type: none"> • Product must be installed on a clean machine—To ensure that all proper changes made by the installation are captured, sequencing needs to be performed on a clean machine (a computer with only the operating system, necessary service packs, and the App-V Sequencer installed on it). A new clean machine would need to be recreated for each application that is sequenced. • Installation directory must be known before sequencing can begin—In order to sequence the application effectively, you must have detailed knowledge of the how the installation is supposed to work. Prior to beginning the sequencing process, you are required to specify the installation directory for the application being sequenced. This information is often not readily available, and may require you to open the installation in an editing tool, such as InstallShield, in order to find it, or to run the installation one time prior to sequencing. 	<p>Instead of installing the package, the Windows Installer Converter obtains the information it needs to create an App-V application directly from the installation. You are not required to have any knowledge of settings within the installation, such as the installation directory. Because there is no need to install the application to obtain this information, no permanent changes are made to the local machine and a clean machine is not required.</p>

Table 9-3 • Benefits of Using the Windows Installer Converter to Create an App-V Application

Windows Installer Converter Benefit	Problem	Solution
Can Test the App-V Application Immediately After Conversion	To run an App-V application, the App-V Client must be installed on the machine. Because sequencing must be performed on a clean machine, which does not have the App-V Client installed, you cannot immediately test a newly-created App-V application on the same machine where you sequenced it.	<p>The Windows Installer Converter includes a launch utility that allows you to launch and test the App-V application locally immediately after conversion before distributing them to the App-V Server.</p>  <p>Note • This feature requires that the App-V Client is installed on the local machine.</p>
Can Perform Bulk Conversion	When using the App-V Sequencer to create App-V packages, you can only sequence one application at a time. For each application that is sequenced, you need to install the application on a clean machine.	You can use the Windows Installer Converter to perform bulk conversions of multiple applications in a directory hierarchy. The Windows Installer Converter has both a user interface and a command line interface.

Creating 64-Bit App-V Packages

AdminStudio 9.5 SP1 supports converting a 64-bit Windows Installer package into a 64-bit Microsoft App-V package, which can be deployed on Windows 64-bit systems with the Microsoft App-V 4.6 64-bit client. This, in combination with our new 64-bit repackaging capability, enables you to convert any 64-bit installation into a 64-bit App-V package.



Note • App-V packages created using earlier versions of AdminStudio are compatible with the new App-V 4.6 client. In some cases, it may be necessary to manually edit the .osd files to specify support for 64-bit operating systems.



Important • It is highly recommended that you perform the conversion of 64-bit Windows Installer packages to App-V packages on a Windows 64-bit machine. If you attempt conversion on a 32-bit Windows machine, it could result in a failure to extract COM information for 64-bit binaries. Also, in some cases, Windows Installer packages contain shortcuts that target executables not found in the package itself. If these shortcuts target executables found in 64-bit Windows folder locations, then these shortcuts will not be handled correctly on 32-bit machines.

The following matrix lists the compatibility of App-V packages created using various versions of AdminStudio with the App-V 4.5 and App-V 4.6 clients:

Table 9-4 • App-V Package Compatibility Matrix

Version of AdminStudio Used to Create App-V Package	Application Type	App-V 4.5 Support	App-V 4.6 (32-bit) Support	App-V 4.6 (64-bit) Support
9.01 or 9.5	32-bit	Yes	Yes	Yes
	64-bit	Not Supported		
9.5 SP1	32-bit	Yes	Yes	Yes
	64-bit	No	No	Yes

Components of an App-V Application

When you use Windows Installer Converter to build an App-V virtual package, the resources you generate are called App-V applications. The components of an App-V application are shown in the following diagram:

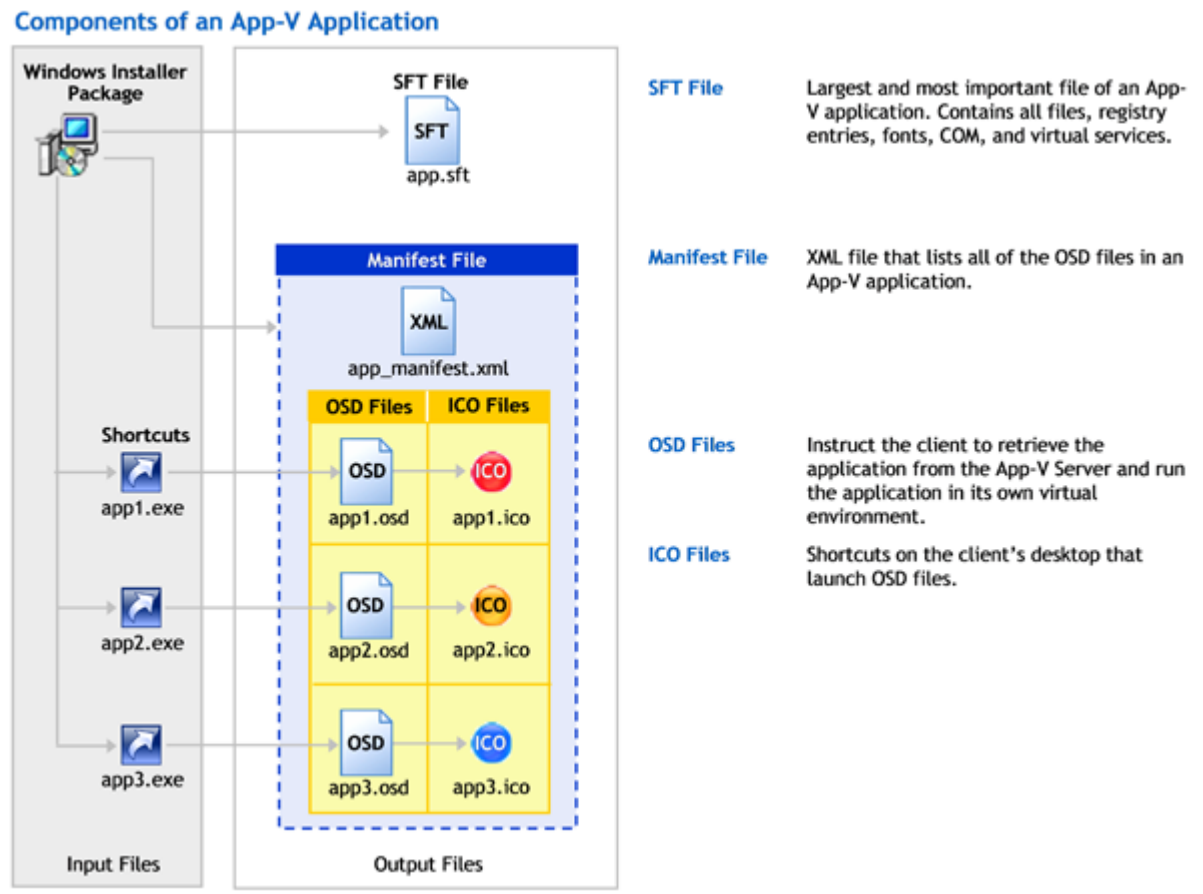



Figure 9-5: Components of an App-V Application

The following table describes the components of an App-V application:

Table 9-5 • Components of an App-V Application

File	Size	Definition
SFT	Up to 4 GB	The largest and most important file in an App-V application. All of the application's assets—including files, registry entries, fonts, COM, and virtual services—reside in this file.

Table 9-5 • Components of an App-V Application

File	Size	Definition
OSD File	2 KB	<p>An XML-based file that instructs the client about the necessary details for retrieving the App-V application from the App-V Server and running the application in its own virtual environment.</p>  <p>Important • Each .osd file needs to identify the published location of its .sft file on the App-V server. You can specify this location on the Package Information page of the InstallShield Microsoft App-V Assistant. However, if you use the Windows Installer Converter to generate an App-V application, you need to use a text editing application, such as Notepad, to individually edit each .osd file and enter the location of the App-V server. See Editing an OSD File to Identify the Location of the App-V Server.</p>
ICO File	24 KB	Shortcuts on the client's desktop that point to an OSD file and launch App-V applications.
Manifest File	2 KB	XML file that lists all of the OSD files in an App-V application.

When you generate an App-V application, its files are saved in a folder named App-VPackage that is created in the same directory as the Windows Installer package you converted. A typical App-V application consists of the following files:

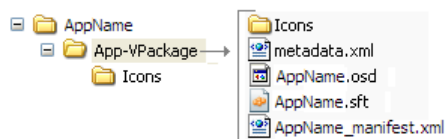


Figure 9-6: App-V Application Files and Directories

Editing an OSD File to Identify the Location of the App-V Server

An .osd file is an XML-based file that instructs the App-V client on how to retrieve the App-V application from the App-V server and how to run it in its virtual environment.

Each .osd file needs to contain the published location of its .sft file on the App-V server. If you set this location on the **Package Information** page of the InstallShield **Microsoft App-V Assistant**, it will automatically be included in all of the .osd files that InstallShield builds for that App-V application. However, if you use the Windows Installer Converter to generate an App-V application, you need to use a text editing application, such as Notepad, to edit each .osd file and enter the published location of its .sft file



Task: *To edit an OSD file by using a text editor:*

1. Open the OSD file using any XML or ASCII text editor—for example, Microsoft Notepad.



Note • Before modifying the .osd file, read the schema prescribed by the .xsd file in the install directory. Failing to follow this schema might introduce errors that prevent a sequenced application from starting successfully.

2. Locate the CODEBASE element. Below is a sample CODEBASE element:

```
<CODEBASE HREF="HTTP://%SFT_SOFTGRIDSERVER%:80/orca.sft" GUID="A895355A-5883-41C6-A144-1BDA12242AAA" PARAMETERS="" FILENAME="{A895355A-5883-41C6-A144-1BDA12242AAA}\Orca.exe" SYSGUARDFILE="{A895355A-5883-41C6-A144-1BDA12242AAA}\osguard.cp" SIZE="2555268"/>
```

3. Locate the HREF attribute of the CODEBASE element and enter a valid URL to the published location of that App-V application's .sft file.



Note • When editing an .osd file, adhere to the prescribed schema and the following guidelines:

- Ensure that named elements are nested within the <SOFTPKG> root element.
- Ensure that element names are in all uppercase letters.
- Be aware that attribute values are case sensitive.
- Type carefully, and observe the XML specifications.

How Windows Services Are Integrated into an App-V Application

When you use the Windows Installer Converter to convert a Windows Installer package to an App-V application, references to Windows services that are encountered are integrated into the App-V application. In a Windows Installer package, a Windows service may be indicated by either an entry in its **ServiceInstall** table or by a Registry entry for Windows services.

- **ServiceInstall table**—If a Windows Installer package's use of a Windows service is indicated by an entry in the **ServiceInstall** table, Windows Installer Converter will convert that entry to a standard Registry entry for Windows services.
- **Registry entry**—If a Windows Installer package's use of a Windows service is indicated by a Registry entry for Windows services (perhaps as the result of being repackaged), Windows Installer Converter does not need to make any changes to support the application's use of the Windows service within the virtual environment.

Start Up and Shut Down Sequences

If an App-V application has an associated Windows service, App-V will start up the Windows service first, in the virtual environment, and then start up the App-V application. You will see the Windows service start up in the Task Manager as a separate process, but App-V will be running the service within the virtual environment.

Upon shut down, App-V will first shut down the App-V application and then shut down the Windows service.

About ThinApp Virtual Packages

You can use the Windows Installer Converter to convert a Windows Installer package to a ThinApp virtual application.

Information about using the Windows Installer Converter is presented in the following topics:

- [About ThinApp Applications](#)
- [Prerequisites for Building a ThinApp Application](#)



Note • You can also convert a Windows Installer package to a ThinApp application using InstallShield Editor's ThinApp Assistant. Using the ThinApp Assistant, you can configure a ThinApp application's Active Directory settings, files, folders, shortcuts, registry settings, isolation options, and build options. See [Getting Started With Application Virtualization](#) and the InstallShield Help Library for information on the ThinApp Assistant.



Note • For information on how to simultaneously build an InstallShield Editor project, a Windows Installer package, and a ThinApp application from your Repackager project, see [Automatically Generating a Virtual Application During Repackager Project Build](#).

About ThinApp Applications

ThinApp is a self-contained application virtualization solution that requires no client-side agents or supporting server infrastructure. A ThinApp application runs within a virtual environment that prevents it from interfering with other software running on the same machine.

ThinApp applications can be deployed on a machine without modifying the local operating system or file system. They run in a “sandbox” (or virtual environment) which protects the local operating system from installation modifications that could affect stability or security. Also, ThinApp applications can be run safely from restricted user accounts without local installation.

Information about ThinApp applications is presented in the following sections:

- [ThinApp Virtual Operating System](#)
- [Benefits of Deploying ThinApp Applications](#)
- [Components of a ThinApp Application](#)

ThinApp Virtual Operating System

A ThinApp application runs in a virtual operating system—a small light-weight component which is embedded with each ThinApp application—that consists of a virtual file system and a virtual registry. When the ThinApp application is run, the virtual operating system environment is merged with the real system environment.

The virtual operating system technology enables entire applications to be packaged into a single .exe file that can be run without an installation process, and without modifying the resident operating system.

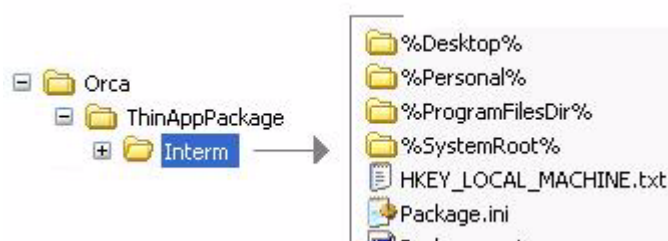
A ThinApp application can be run from a network or offline on the local machine.

Components of a ThinApp Application

When you use Windows Installer Converter, Repackager, or InstallShield Editor to build a ThinApp virtual package, the resources you generate are called **ThinApp applications**.

When package conversion is complete, a **Conversion completed** message appears in the **Output** window and the path to the generated ThinApp application is listed, such as:

C:\AdminStudio Shared\My Application\ThinAppPackage



The ThinApp application is created in a folder named ThinAppPackage that is created in the same directory as the Windows Installer package you converted.

The number of files included in a ThinApp application depends upon how many shortcuts are defined:

Table 9-6 • Components of a ThinApp Application






Number of Shortcuts	ThinApp Application Components	Description
1 shortcut	<p>ProductName.exe</p>  <p>AdminMaster70.EXE AdminMaster IT Toolz, Inc.</p>	<p>The ThinApp application consists of a single executable (.exe) file:</p> <ul style="list-style-type: none">• Launching the application—This executable file is used to launch the ThinApp application.• Location of application data—This executable file contains all of the files, registry keys, DLLs, ThinApp components, and third party libraries that are required for the application to run.

Table 9-6 • Components of a ThinApp Application

Number of Shortcuts	ThinApp Application Components	Description
More than 1 shortcut	ProductName.exe FeatureName.exe Package.DAT  XYZPhotoBrowse40.EXE XYZ Photo Browse XYZ Software, Inc.  XYZPhotoTouchUp40.EXE XYZ Photo TouchUp XYZ Software, Inc.  Package.DAT DAT File 128,253 KB	The ThinApp application consists of two or more executable files and a Package.DAT file: <ul style="list-style-type: none"> • Launching the application—Each of the executables is used to launch the ThinApp application or a specific feature of the ThinApp application. • Location of application data—The Package.DAT file contains all of the files, registry keys, DLLs, ThinApp components, and third party libraries that are required for the application to run.
Metadata File	metadata.xml	An XML file created during AdminStudio 9.0+ package conversion that contains metadata identifying the original Windows Installer package that was used to create the virtual package.  Note • Because of this file, you are able to import this virtual package into the Application Catalog and associate it with its source Windows Installer package. For more information, see Importing Virtual Packages .



Caution • Modifying these files directly is **not recommended**. To make any modifications, use the InstallShield ThinApp Assistant.

Intermediate Data Files: Interm Directory

When a ThinApp application is built, files that support the ThinApp application build process are extracted out of the Windows Installer package and saved in a subdirectory of the ThinAppPackage directory named the Interm directory.



Figure 9-7: Interm Subdirectory of the ThinAppPackage Directory

The data in this directory is then compiled into ThinApp application as part of the build process. The data in the Interm directory *does not* need to be distributed with the ThinApp application.

Benefits of Deploying ThinApp Applications

Deploying ThinApp applications provides the following benefits:

- **Reduces time to deployment and costs associated with testing**—Applications can be deployed and run in independent sandboxes, eliminating the need for expensive and time-consuming multi-application regression testing. This reduces the time to deployment and the costs associated with testing.
- **Fast, lightweight virtualization** —ThinApp does not use emulation, so all processes are executed natively at full speed.
- **Reduces the cost of maintaining secure locked-down desktops**—ThinApp applications can run in restricted user accounts without requiring any host modifications.
- **Enhances work-force mobility, business continuity and disaster recovery**—ThinApp applications can be run off-line, directly from any external media including USB Flash, CD-ROM, and off-line laptops.
- **No infrastructure changes needed**—ThinApp applications can be deployed using any existing software deployment systems including Active Directory and SMS. ThinApp has no client or server components to manage or maintain and ThinApp can transparently stream large applications from any network attached storage devices without server software.
- **Sandboxing prevents modifications**—ThinApp redirects all changes intended for the host computer's file system and registry to a private per-user sandbox. Sandboxes can be located on a network share, allowing application settings to follow users as they move from machine to machine. For mobile users, sandboxes can be stored on local USB flash drives, thus preventing damage to the host computer or accidental host storage of sensitive data.

Prerequisites for Building a ThinApp Application

AdminStudio will convert the package installation into a format compatible with ThinApp. However, the ThinApp build process requires the availability of certain ThinApp tools.

As a prerequisite to building a ThinApp application from AdminStudio, you must have installed ThinApp **and accepted any and all license agreements**.



Caution • If you install ThinApp but you have not yet accepted the license agreement, the build process will fail. For more information, see the [VMware Web site](#).

About Citrix Virtual Packages

You can use the Windows Installer Converter to convert a Windows Installer package to a Citrix profile for deployment on Citrix XenApp.

Information about using the Windows Installer Converter is presented in the following topics:

- [About Citrix XenApp and Citrix Profiles](#)

- Benefits of Deploying Citrix Profiles



Note • You can also convert a Windows Installer package to a Citrix profile using InstallShield Editor's Citrix Assistant. Using the Citrix Assistant, you can configure a Citrix profile's operating system and language requirements, files, folders, shortcuts, registry settings, script execution, isolation options, and build options. See [Getting Started With Application Virtualization](#) and the InstallShield Help Library for information on the Citrix Assistant.



Note • For information on how to simultaneously build an InstallShield Editor project, a Windows Installer package, and a Citrix profile from your Repackager project, see [Automatically Generating a Virtual Application During Repackager Project Build](#).

About Citrix XenApp and Citrix Profiles

Citrix XenApp is an application delivery system for Windows applications. When you use Repackager or InstallShield Editor to prepare a Windows Installer package for deployment on Citrix XenApp, the resources you generate are called **profiles**.

Overview information about Citrix XenApp and Citrix profiles is presented in the following topics:

Table 9-7 • Overview of Citrix XenApp

Topic	Description
About Citrix XenApp	Provides an overview of how Citrix XenApp works and provides a diagram illustrating application delivery.
About Citrix Profiles (.profile)	Lists the files and directories that comprise a Citrix profile.

About Citrix XenApp

Citrix XenApp is an application delivery system for Windows applications that offers both application virtualization and application streaming. Applications are centralized on Citrix XenApp and then those applications are deployed to users throughout the enterprise. These deployed applications run within isolation environments that prevent them from interfering with other software running on the same machine.

Citrix XenApp: 2 Steps to Application Delivery

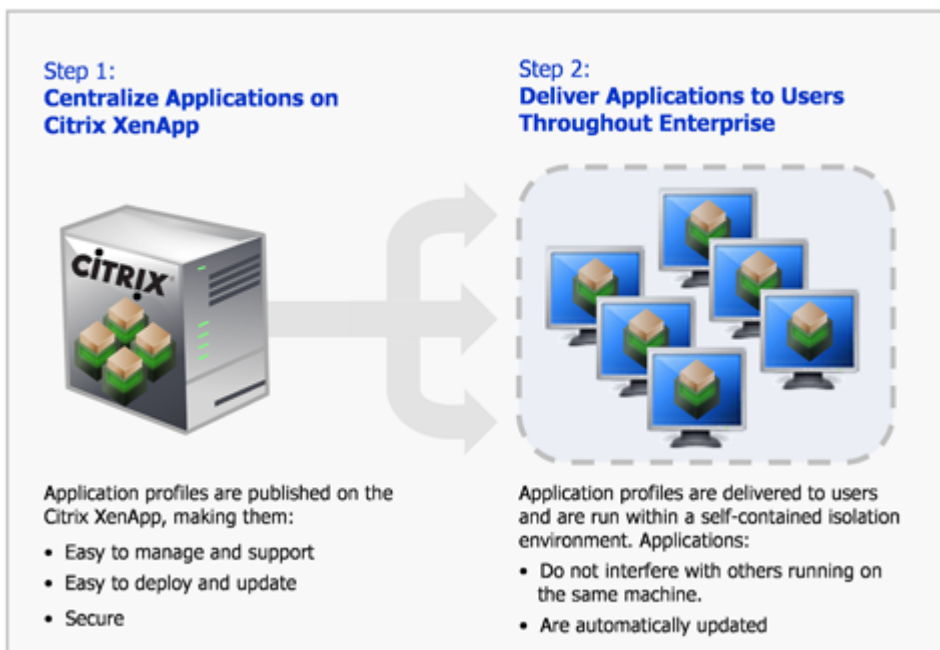


Figure 9-8: Citrix XenApp: Two Steps to Application Delivery

When applications are deployed on a Citrix XenApp, users can run those applications in an isolation environment, without installing, while connected or offline. Applications behave just like they were installed locally, but without any of the problems of installation, such as interfering with other applications on the same device. Files are saved locally and individual settings are preserved. Every time the application is run, it checks for errors or updates and they are delivered automatically.



Note • For more information, see [Benefits of Deploying Citrix Profiles](#).

About Citrix Profiles (.profile)

When you use Windows Installer Converter, Repackager, or InstallShield Editor to prepare a Windows Installer package for deployment on Citrix XenApp, the resources you generate are called **profiles**.

When package conversion is complete, the Windows Installer Converter, Repackager, or InstallShield Editor displays the path to the generated virtual package, such as:

```
C:\AdminStudio Shared\MyPackage\CitrixProfile\MyPackage.profile
```


These files are saved in a subfolder of a folder named `CitrixProfile` that is created in the same directory as the Windows Installer package you converted. The profile, which is published on Citrix XenApp, consists of the following:



Figure 9-9: Profile Files and Directories

A profile contains the following files and directories:

Table 9-8 • Components of an Application Profile

Component	Name	Description
Profile Manifest File	myapp.profile	An XML file that defines the profile.
CAB File	[alphanumeric_string].cab	Compressed cabinet file that provides the isolation environment contents for the application.
Hashes File	Hashes.txt	Hash key file for digital signatures and signing profiles.
Icons File	Icons.bin	Icons repository.
Scripts Folder	Scripts	Folder containing any pre- launch or post-exit scripts that you have chosen to include.
Metadata File	metadata.xml	<p>An XML file created during AdminStudio 9.0+ package conversion that contains metadata identifying the original Windows Installer package that was used to create the virtual package.</p> <p> Note • Because of this file, you are able to import this virtual package into the Application Catalog and associate it with its source Windows Installer package. For more information, see Importing Virtual Packages.</p>



Caution • Modifying these files directly is **not recommended**. To make any modifications, use the InstallShield Citrix Assistant.

A profile can contain a single application or suite of applications. For more information, see:

- [Converting a Single Application to a Virtual Package](#)

- [Converting a Suite of Applications into a Virtual Package](#)
- [Converting a Directory of Packages into Individual Virtual Packages](#)

Benefits of Deploying Citrix Profiles

Converting a Windows Installer package to a Citrix profile and deploying it on a Citrix XenApp offers the following benefits:

- [Reduces Application Conflicts](#)
- [Enables Rapid, Low Cost Application Deployment](#)
- [Enables Automatic Software Updates](#)
- [Centralized Application Management Provides Controlled Access and Security](#)
- [Enables User-Based Application Access Rather Than Machine-Based Access](#)

Reduces Application Conflicts

Traditionally to deploy an application throughout an enterprise, the application was installed on each user's desktop. Therefore, prior to installation, each application had to be tested for conflicts against each target desktop image (operating system with existing applications). After resolving conflicts that were found during testing, each application then had to be installed on each desktop. This process was very time consuming not only during initial installation, but also when applying patches or upgrading.

Citrix profiles run within isolation environments, which separate the interaction between an application and the underlying operating system's resources in order to prevent the applications from interfering with others running on the same machine. Because applications do not interact, the need to perform any conflict analysis and regression testing prior to deployment is eliminated. This not only results in rapid application deployment, but it also reduces the total cost of application delivery, due to decreased labor by IT.

Also, because users running applications in an isolation environment encounter no conflicts with other applications, user calls to the help desk are decreased.

Enables Rapid, Low Cost Application Deployment

Deploying Citrix profiles on Citrix XenApp simplifies the deployment of new applications, updates and patch deployment, regardless of the diversity of the access devices, software languages, computing architectures, and networks that are involved.

- **Only a single instance of the application is installed**—Instead of deploying, managing, updating and securing a vast array of heterogeneous client software on each individual user's access device, a single instance of the application is installed on Citrix XenApp. The IT department only has to test for one environment, and deploy and update in one place. This reduces the cost of application installation and support. Also, you can deploy a Citrix profile once on a Citrix XenApp and replicate it to other Citrix XenApps within the existing enterprise infrastructure.
- **Prevents application-specific server silos**—Deploying applications on Citrix XenApp prevents the build-up of application-specific server silos because you can safely install and reliably run multiple application versions and incompatible applications on the same server.

- **Enables you to quickly install and update software throughout your enterprise**—Because you can manage the delivery of all of your Windows-based applications from one centralized location, there is no need to go from desktop to desktop, travel from office to office, or wait for laptops to return to headquarters in order to install or update software. With Citrix XenApp, you can deliver applications and updates instantly anywhere, any time—to offshore employees, outsourcers, new branch offices, new mergers and acquisitions, and mobile workforces.

Enables Automatic Software Updates

When an upgrade or patch needs to be deployed, you would only need to update the Citrix profile on Citrix XenApp, which will then automatically update all of the instances of that Citrix profile throughout the enterprise. This means that users always have the latest application updates and patches, automatically.

Centralized Application Management Provides Controlled Access and Security

With Citrix XenApp, you can centralize applications and data in secure data centers, which increases data security and ensures fast, reliable performance. Centralized application management using Citrix XenApp provides the following benefits:

- **Enhances security**—Enables you to control, protect, and retain intellectual property centrally to reduce the chance for data loss and theft. Citrix XenApp helps you prevent data from leaving the data center without your explicit permission, which supports regulatory compliance and security objectives. You can provide authorized access to appropriate users—such as employees, customers, and partners—while verifying the ongoing security of the environment.
- **Can provide managed access to applications to users outside of your organization**—You can standardize the use of applications, without having to standardize the machines that the applications use. This enables you to provide managed access to applications from computers that are not your own corporate assets, such as from contractor or consultant computers.
- **Monitors application usage and performance**—Citrix XenApp gives you end-to-end visibility into application usage and performance. It gives IT administrators the power to understand who is using what, how often, and to what extent. They can observe, monitor, measure, audit, report and archive all the dimensions of information flow throughout the computing environment. This enables informed decisions regarding application consolidation and retirement, capacity planning, service level agreements and departmental charge-back.
- **Enables identity-driven access**—Citrix XenApp enables you to provide identity-driven access tailored to any user environment. It automatically analyzes the user's permissions and then delivers the appropriate level of access to applications without compromising security. Depending on who and where users are and what device and network they're using, they may be granted different levels of access. You can also easily “decommission” applications by simply turning off a user's permission to it.

Enables User-Based Application Access Rather Than Machine-Based Access

Users can access their applications anywhere on the network, regardless of where they are or what device they are using.

Converting a Single Application to a Virtual Package

This section explains how to use the Windows Installer Converter to convert a single Windows Installer package into a virtual package.



Task: *To use the Windows Installer Converter to convert a single application to a virtual package:*

1. On the Repackager Home page, click **Convert a Windows Installer Package to an Application Virtualization Format**. The **Welcome Panel** of the Windows Installer Converter opens.



Note • You can also open the Windows Installer Converter by selecting **Windows Installer Converter** on the **Tools** menu.

2. Click **Next** to continue. The **Select Conversion Target** panel opens.
3. Select **Microsoft App-V Application**, **ThinApp Application** and/or **Citrix Profile** and click **Next**. The **Conversion Mode** panel opens.
4. Select **Single Application** and click **Next**. The **Select Windows Installer Package** panel opens.



Note • For information on performing Pre-Conversion Evaluation, see [Testing Windows Installer Packages for Virtualization Compatibility](#).

5. Click **Browse** and select the Windows Installer package that you want to convert to a virtual package. All of the transform files that are in the same folder as the selected Windows Installer package are automatically listed in the **Windows Installer Transform Files (.MST)** list.



Caution • It is preferable that all of the transforms that you include in a virtual package be located in the same folder as the Windows Installer .msi package so that they can be accessed when the virtual package is built. Only those transform files that are in this folder will be listed on the **Transform Files (.MST)** list on this panel.

6. Select those transform files that you would like to include in the virtual package.
 - The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up (↑) and Move Down (↓) buttons.
 - Use the Delete button (✖) to delete a transform from the list.
7. Click **Next**. The **Edit Windows Installer Package** panel opens, where you are given the opportunity to use the InstallShield Editor Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant to edit this Windows Installer package before creating the virtual package.

With the Microsoft App-V Assistant you can:

- Enter the package name and root folder name.
- Set operating system requirements.
- Identify the deployment server.
- Modify package contents, registry settings, and shortcuts.
- Set isolation options for selected files, folders, and/or registry keys.
- Control virtual application interaction between multiple App-V packages.

With the ThinApp Assistant you can:

- Modify package contents, registry settings, and shortcuts.
- Set custom isolation options on folders and registry entries.
- Set operating system and/or language requirements.
- Control access to the application using Active Directory.
- Compress the ThinApp application files.

With the Citrix Assistant you can:

- Modify package contents, registry settings, and shortcuts.
- Set custom isolation options on files, folders, and registry entries.
- Set operating system and/or language requirements.
- Specify whether to digitally sign the virtual package.



Important • If you are using standalone Repackager or if you have not installed InstallShield Editor on this machine, the **Edit Windows Installer Package** panel will not be displayed.

8. Do one of the following:

- a. To open the selected package in InstallShield Editor**, click the **Edit Using InstallShield Editor** button. The Windows Installer Converter wizard will close, and the package will open in InstallShield Editor in either Direct Edit Mode or in Direct MST Mode (if any transforms have been selected).

Then continue with the instructions in *Creating Customized Virtual Applications* in the InstallShield Help Library.



Note • If you have selected transform files to be applied to this package, any edits you make to this Windows Installer package in InstallShield Editor will be saved in a separate transform file named *MSIConverterSettings.mst*, which will be located in the same folder as the selected package.

- b. If you don't want to open the selected package in InstallShield Editor**, continue with the following step.

9. To build the virtual package, click **Convert**.

The **Package Conversion** panel opens and conversion begins. During the conversion process, status messages are displayed in the **Output** window.

Error and warning messages are listed in the **Error List**. You can click on the error number to view a help topic that includes information on how to resolve that error, if possible. For a complete list of error and warning messages, see [Virtualization Conversion Error Messages](#).



Note • The Windows Installer Converter ignores services and user-defined custom actions. For more information, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

10. Review the virtual package files.
 - For information on the files that comprise an App-V application, see [Components of an App-V Application](#).
 - For information on the files that comprise a Citrix profile, see [About Citrix Profiles \(.profile\)](#).
 - When distributing the ThinApp application, be sure to include **all** of its component files. For information on the files that comprise a ThinApp application, see [Components of a ThinApp Application](#).
11. Click **Finish** to close the Windows Installer Converter. The virtual package is ready for deployment.

Converting a Suite of Applications into a Virtual Package

If you want to convert multiple Windows Installer packages, perhaps a group of applications that need to work together, into one virtual package, select the **Virtual Suite** option on the **Conversion Mode** panel of the Windows Installer Converter.



Task: *To use the Windows Installer Converter to convert a suite of applications to a virtual package:*

1. On the Repackager Home page, click **Convert a Windows Installer Package to an Application Virtualization Format**. The **Welcome Panel** of the Windows Installer Converter opens.







Note • You can also open the Windows Installer Converter by selecting **Windows Installer Converter** on the **Tools** menu.

2. Click **Next** to continue. The **Select Conversion Target** panel opens.
3. Select **Microsoft App-V Application**, **ThinApp Application** and/or **Citrix Profile** and click **Next**. The **Conversion Mode** panel opens.
4. Select **Virtual Suite** and click **Next**. The **Select a Suite of Packages** panel opens.



Note • Because you have chosen to create a Virtual Suite of applications, the **Evaluate package(s) for virtualization suitability** option is disabled. For information on performing Pre-Conversion Evaluation, see [Testing Windows Installer Packages for Virtualization Compatibility](#).

5. Click **Browse** and select the primary package in the virtual suite—the application that end users will primarily interact with.
6. Click the New button () and select the Windows Installer packages that are often used indirectly by the primary package, such as driver files, client components, etc. After each file is selected, it will be listed in the **Additional Suite Windows Installer Files (.MSI)** list.
 - The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.
 - Use the Delete button () to delete a package from the list.
7. To build the virtual suite of packages, click **Convert**.

The **Package Conversion** panel opens and conversion begins. During the conversion process, status messages are displayed in the **Output** window.

Error and warning messages are listed in the **Error List**. You can click on the error number to view a help topic that includes information on how to resolve that error, if possible. For a complete list of error and warning messages, see [Virtualization Conversion Error Messages](#).

When conversion is complete, a link to a log file and the location of the virtual packages appears in the **Output** window.



Note • The Windows Installer Converter ignores services and user-defined custom actions. For more information, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

8. Review the virtual suite package files.
 - For information on the files that comprise an App-V Application, see [Components of an App-V Application](#).
 - When distributing the ThinApp application, be sure to include **all** of its component files. For information on the files that comprise a ThinApp application, see [Components of a ThinApp Application](#).
 - For information on the files that comprise a Citrix profile, see [About Citrix Profiles \(.profile\)](#).



Note • The creation of application shortcuts varies depending upon which virtual application format you are using:

- **Microsoft App-V**—When creating a Microsoft App-V application, Windows Installer Converter converts the installation data from the original Windows Installer package into an `.sft` file that is compatible with Microsoft App-V version 4.5. The installation shortcuts are converted into `.osd` files, which provide access to programs within the `.sft` file.

- **ThinApp**—When creating a virtual suite of packages using ThinApp, Windows Installer Converter creates a program shortcut for every application in the suite and all shortcuts are presented to end users.
- **Citrix**—When creating a virtual suite of packages using Citrix XenApp, Windows Installer Converter also creates a program shortcut for every application in the suite. However, when you distribute a virtual package using Citrix XenApp, you can only select one shortcut. If you want this virtual suite to have more than one shortcut, you will need to distribute it more than once.

9. Click **Finish** to close the Windows Installer Converter.

Converting a Directory of Packages into Individual Virtual Packages



Edition • The **Batch Convert** option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.

This section explains how to use the Windows Installer Converter to convert a directory of Windows Installer packages into individual virtual packages. It is recommended that each of these Windows Installer packages be in a separate subdirectory of the selected directory.



Note • For more information, see [Batch Conversion](#).



Task:

To use the Windows Installer Converter to convert a directory of packages to individual virtual packages:

1. On the Repackager Home page, click **Convert a Windows Installer Package to an Application Virtualization Format**. The **Welcome Panel** of the Windows Installer Converter opens.



Note • You can also open the Windows Installer Converter by selecting **Windows Installer Converter** on the **Tools** menu.

2. Click **Next** to continue. The **Select Conversion Target** panel opens.
3. Select **Microsoft App-V Application**, **ThinApp Application** and/or **Citrix Profile** and click **Next**. The **Conversion Mode** panel opens.



Note • If you select the **Evaluate package(s) for virtualization suitability** option on the **Conversion Mode** panel, the package(s) is tested prior to conversion. After viewing the test results, you can choose the packages you want to convert. See [Testing Windows Installer Packages for Virtualization Compatibility](#).

4. Select **Batch Convert** and click **Next**. The **Select a Directory** panel opens.



Note • For information on performing Pre-Conversion Evaluation, see [Testing Windows Installer Packages for Virtualization Compatibility](#).

5. Click **Browse** and select the directory containing the Windows Installer packages that you want to convert to individual virtual packages. It is recommended that each of these Windows Installer packages be in a separate subdirectory of the selected directory.
6. Click **Convert**. The **Package Conversion** panel opens and conversion begins. During the conversion process, status messages are displayed in the **Output** window.

Error and warning messages are listed in the **Error List**. You can click on the error number to view a help topic that includes information on how to resolve that error, if possible. For a complete list of error and warning messages, see [Virtualization Conversion Error Messages](#).



Note • The Windows Installer Converter ignores services and user-defined custom actions. For more information, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

When conversion is complete, a link to a log file and the location of the virtual packages appears in the **Output** window.

7. Review the virtual package files.
 - For information on the files that comprise an App-V Application, see [Components of an App-V Application](#).
 - When distributing the ThinApp application, be sure to include **all** of its component files. For information on the files that comprise a ThinApp application, see [Components of a ThinApp Application](#).
 - For information on the files that comprise a Citrix profile, see [About Citrix Profiles \(.profile\)](#).
8. Click **Finish** to close the Windows Installer Converter. The virtual packages are ready for deployment.

Testing Windows Installer Packages for Virtualization Compatibility

You can use the Pre-Conversion Evaluation feature of Windows Installer Converter to test applications for virtualization compatibility before proceeding with conversion. This feature gives you a way to evaluate at a high level how easily a virtualization solution could be adopted.

When performing a pre-conversion evaluation, Windows Installer Converter applies a set of pre-defined rules to evaluate packages. However, Windows Installer Converter also incorporates an extensible test framework that enables you to write your own custom rules to detect and possibly fix issues that could prevent the successful creation of a virtual package. As you gain experience in virtualization conversion, you can build additional rules to address any issues that you have found, making the virtualization experience in the future even more streamlined for your organization.

Information about performing pre-conversion testing is presented in the following topics:

- [Using Windows Installer Converter to Test Packages for Virtualization Compatibility](#)
- [About Rules and Tests](#)
- [Pre-Conversion Evaluation Rules Shipped With AdminStudio](#)
- [Extensible Test Framework: Writing Your Own Rules for Virtualization Conversion](#)

Using Windows Installer Converter to Test Packages for Virtualization Compatibility

You can use the Pre-Conversion Evaluation feature of Windows Installer Converter to test applications for virtualization compatibility before proceeding with conversion. This feature gives you a way to evaluate at a high level how easily a virtualization solution could be adopted.

Test results are displayed in HTML format, which can be saved and reviewed by others. After reviewing the test results you can decide which packages you want to convert. Those that are deemed suitable for conversion will be pre-selected. You can then decide to proceed with conversion of the selected packages or wait until a later date to perform conversion.



Task:

To test Windows Installer packages for virtualization compatibility:

1. Open the Windows Installer Converter by doing one of the following:
 - On the Repackager **Home** page, select **Convert a Windows Installer Package to an Application Virtualization Format**.
 - On the Repackager **Tools** menu, select **Windows Installer Converter**.
 - Open the Windows **Start Menu** and select **Windows Installer Converter** under **All Programs/AdminStudio/AdminStudio 9.5 Tools/Virtualization**.

The **Welcome Panel** of the Windows Installer Converter opens.

2. Click **Next** to continue. The **Select Conversion Target** panel opens.
3. Select the virtualization format(s) that you are planning to convert to. The **Conversion Mode** panel opens.
4. Select your desired conversion mode:
 - Single Application
 - Batch Convert

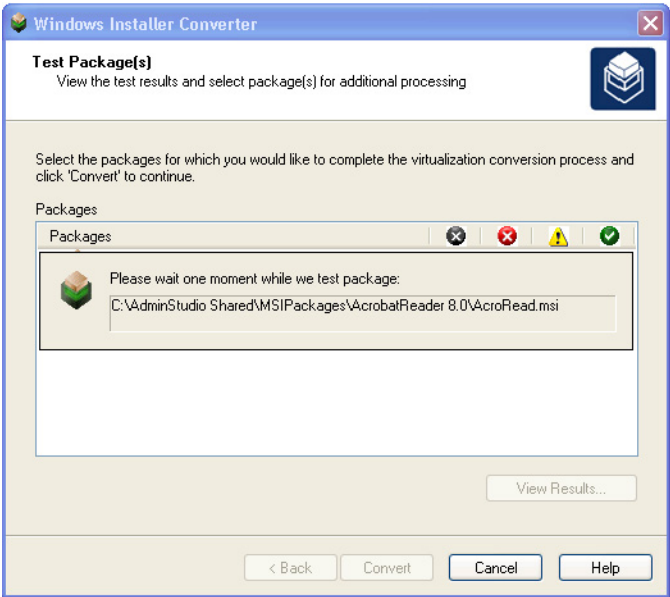
These conversion modes are explained in [Types of Virtual Package Conversion](#).



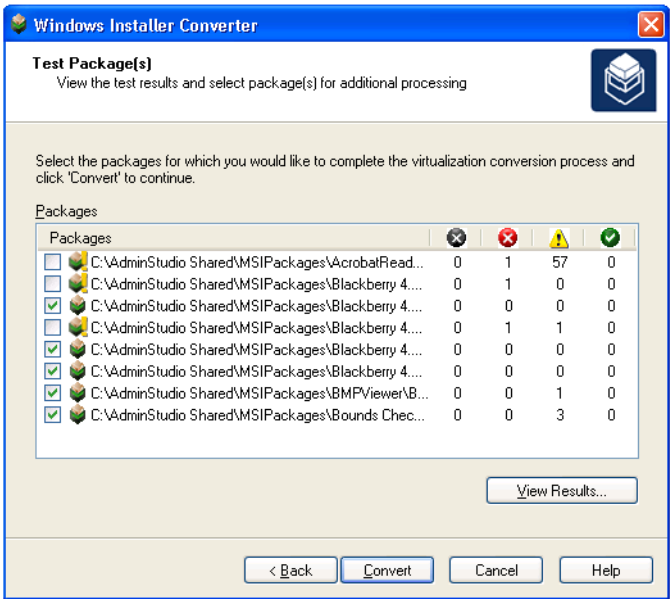
Note • If you select the **Virtual Suite** conversion mode, the testing option is not supported.

5. Select the **Evaluate package(s) for virtualization suitability** option.

- Proceed with package conversion as described in [Converting a Single Application to a Virtual Package](#), [Converting a Suite of Applications into a Virtual Package](#), or [Converting a Directory of Packages into Individual Virtual Packages](#).
- Click **Test**. The **Test Package(s)** panel opens and testing begins.







When testing is complete, the results for each package are displayed:



Chapter 9: Converting Windows Installer Packages to Virtual Applications

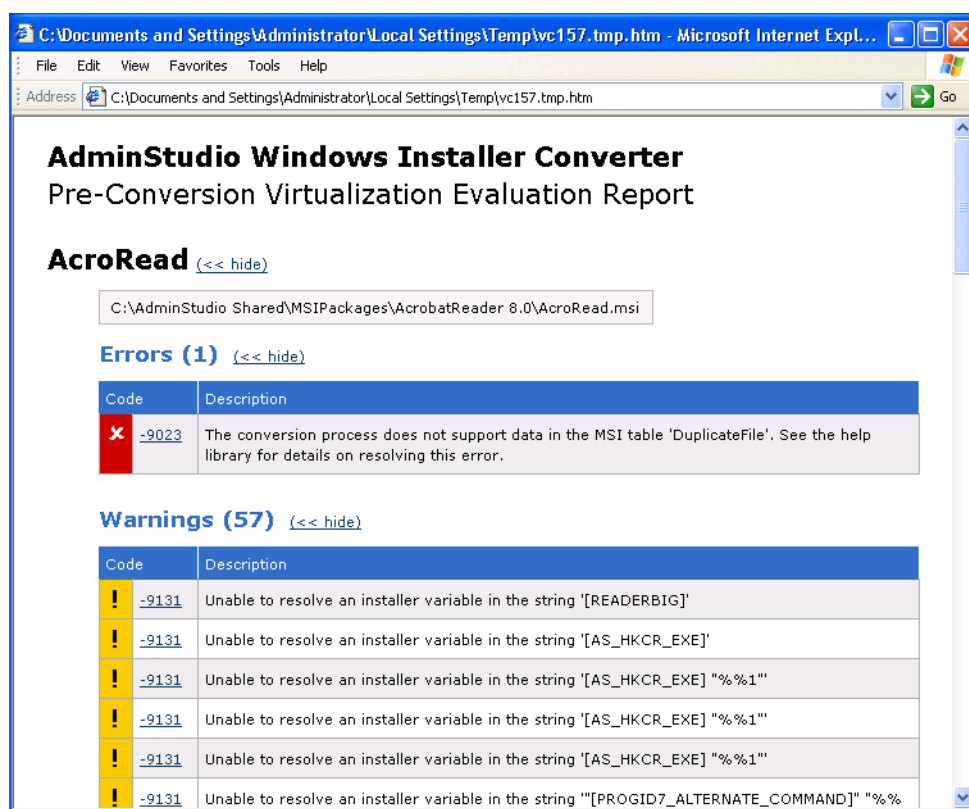
Testing Windows Installer Packages for Virtualization Compatibility

The number of errors and warnings generated are displayed under the appropriate icon:

Icon	Name	Description
	Fix	Windows Installer Converter discovered an issue and resolved it.
	Warning	The virtual package will probably run, but may not.
	Error	The virtual package probably won't run, but under some circumstances it may.
	Fatal Error	The virtual package will not run.

Those packages that did not have any errors or fatal errors are selected and the **Convert** button is enabled.

- Click **View Results**. An HTML report is generated that lists all of the messages and provides links to a help topic for each message:



- To save the HTML report, select **Save As** from the **File** menu.
- Select those packages that you want to convert and click **Convert**. Conversion begins.

When package conversion is complete, a **Conversion completed** message appears in the **Output** window and the path to the generated virtual package(s) is provided.

11. Click **Finish** to close the Windows Installer Converter.

About Rules and Tests

When performing a pre-conversion evaluation, Windows Installer Converter generates messages from two sources:

- **Generated by the Microsoft App-V, ThinApp, and/or Citrix build engines**—Error and Warning messages are generated by the virtual package build engines that identify potential problems in the conversion process.
- **Generated by running tests using XML-based rules files**—Fatal Error, Error, Warning, and Fix messages are generated by Windows Installer Converter by applying both pre-defined and user-supplied XML-based rules that identify (and possibly fix) problems that have been discovered when converting specific applications and specific components under certain conditions.

Therefore, the testing capability of Windows Installer Converter can be significantly increased as you develop and share additional XML-based rules. The more *rules* that you have, the more *tests* Windows Installer Converter can run, and the more *fixes* it can make.

Pre-Conversion Evaluation Rules Shipped With AdminStudio

When performing a pre-conversion evaluation, the Windows Installer Converter applies the rules listed in the following table. These rules are defined in the AdminStudio Default.xml file.

Table 9-9 • Windows Installer Converter Pre-Conversion Evaluation Rules

Number	Type	Rule Name	Description
11000	Fix	Exclude TCP/IP Registry Entries From Citrix Packages	Excludes TCP/IP registry entries from all Citrix packages.
11001	Fatal	VMware Fatal Error	VMware cannot be virtualized. Warning is displayed stating that virtualization conversion will fail when the application being virtualized is VMWare.
11003	Warning	Control Panel Applet / Citrix	Warning is displayed when a Citrix application contains a control panel applet.
11004	Fix	Create Control Panel Applet Shortcut for ThinApp	Creates a default shortcut for ThinApp Control Panel Applets.

Table 9-9 • Windows Installer Converter Pre-Conversion Evaluation Rules

Number	Type	Rule Name	Description
11005	Fatal	Quicktime 7.4.1 Error	Warning is displayed stating that QuickTime 7.4.1 is known to have errors when running from a virtual package, and recommends that you use QuickTime 7.4.5 instead.
11006	Fix	Exclude AdobePDFSettings Registry Settings	Excludes the AdobePDFSettings registry key. Adobe Distiller will recreate these settings on first use.
11007	Fix	Exclude URL Shortcut	Windows Installer Converter will exclude the shortcut to the .URL file. App-V does not launch these files properly.

Extensible Test Framework: Writing Your Own Rules for Virtualization Conversion

AdminStudio has an extensible test framework for the Windows Installer Converter that you can use to write your own custom rules to evaluate virtual packages.

- **Rules detect specific issues**—Rules should be designed to detect specific issues in specific packages.
- **Rules are run during Pre-Conversion Evaluation**—The existing Windows Installer Converter Pre-Conversion Evaluation feature will also run these tests, so any test that detects an error on conversion will also be able to detect it during a pre-conversion evaluation.
- **Tests can contain fixes that are implemented during conversion**—Tests will sometimes contain remediations that can correct a failure automatically. The remediations are only run on conversion and not during a pre-conversion evaluation.

About Custom Rules

A rule takes some action if some condition is met. For example, if packaging Office 2007, fully isolate registry keys shared with Office 2003 so that these applications can exist side-by-side.

Benefits of Using Custom Rules

The benefits of writing custom rules are:

- You are documenting knowledge gained while packaging an application.
- You gain the ability to share that knowledge with others.
- When building a virtual package, you can automatically execute the rule's fix to address the issue.

Execution of Rules

Rules get evaluated and executed when building a virtual package. Rules also get executed when a Pre-Conversion Evaluation is performed.

Components of a Rule

Rules are defined completely in XML format. The Windows Installer Converter Rules Engine will load any *.xml file in the C:\Program Files\AdminStudio\9.5\Repackager\System\Rules folder that contains the root element of <AdminStudioRules>.

Rules are made up of two main items: a Fingerprint and an Action.

- **Fingerprint**—The condition associated with the Rule.
- **Action**—The action that will be taken when the Fingerprint condition has been satisfied.

Here is a very simple sample rule:

Table 9-10 • Sample Custom Rule for Windows Installer Converter Rules Engine

```
<Rule title="Adobe Distiller Exclude AdobePDFSettings"
description="Exclude AdobePDFSettings registry settings"
author="AdminStudio.Default" provider="Any">
  <Fingerprint name="Adobe Distiller AdobePDFSettings" description="">
    <Registry
      hive="HKEY_CURRENT_USER" key="Software\Adobe\Acrobat Distiller\AdobePDFSettings\*" />
    </Fingerprint>
  <Action description="Exclude AdobePDFSettings">
    <Message id="11006" text="Excluding the registry key Software\Adobe\
      Acrobat Distiller\AdobePDFSettings. Adobe Distiller will recreate these settings on first
      use." severity="Fix"/>
    <Exclude>
      <Registry hive="HKEY_CURRENT_USER" key="Software\Adobe\
        Acrobat Distiller\AdobePDFSettings\*" />
    </Exclude>
  </Action>
</Rule>
```

In this example, the following Fingerprint and Action are specified:

Table 9-11 • Example of Fingerprint and Action in Sample Custom Rule

Item	Description
Fingerprint	The Fingerprint, which is named <code>Adobe Distiller AdobePDFSettings</code> , searches the virtual packages for a Registry key of <code>Software\Adobe\Acrobat Distiller\AdobePDFSettings*</code> in the <code>HKEY_CURRENT_USER</code> Registry hive.

Table 9-11 • Example of Fingerprint and Action in Sample Custom Rule

Item	Description
Action	If the Fingerprint condition is met, an Action named <code>Exclude AdobePDFSettings</code> will be implemented, which excludes the Registry key of <code>Software\Adobe\Acrobat Distiller\AdobePDFSettings*</code> in the <code>HKEY_CURRENT_USER</code> Registry hive for that application.

Creating Your First Custom Rule

To create a custom rule for use in virtual package testing and conversion using Windows Installer Converter, perform the following steps. These instructions assume that you are familiar with some of the basic concepts of XML.

To ensure that as new versions of `AdminStudio.Default.xml` are released, your custom rules are not overwritten, it is recommended that you add your custom rules to your own XML file and not to the `AdminStudio.Default.xml` file. You should create one XML file for each application or a common set of applications.



Tip • The rule XML definition has a schema file. If you use an editor that is schema-aware, it can significantly improve the ease with which rule files are created.



Task: To create a custom rule:

1. Create an XML file in the `C:\Program Files\AdminStudio\9.5\Repackager\System\Rules` folder called `MyRules.xml` and define the following root element:

```
<?xml version="1.0" encoding="utf-8"?>
<AdminStudioRules xmlns="http://adminstudio.com/AdminStudio.Rules.xsd"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://adminstudio.com/AdminStudio.Rules.xsd AdminStudio.Rules.xsd">
  </AdminStudioRules>
```

2. Add a `Rule` element as a child element of `AdminStudioRules`. This element is the basic definition for your first rule. If you wanted to define multiple rules, you would simply create multiple `Rule` elements.

```
<Rule title="My Rule Title" description="My Rule Description" author="Author" provider="Any">
  </Rule>
```

3. Add a `Fingerprint` element as a child element to the rule. A `Fingerprint` is the condition associated with the rule. First add an empty `Fingerprint`. When a `Fingerprint` is empty it will always be satisfied and thus the rule will always be applied to every package you build.

```
<Fingerprint name="Always Satisfied" description="">
  </Fingerprint>
```

4. Add an `Action` element as a child element to the rule. An `Action` is what happens when the rule `Fingerprint` is satisfied.

```
<Action description="My First Action">
  <Message id="12000" text="I will automate my knowledge" severity="Warning"/>
```

```
</Action>
```

5. Save the XML file. The final results should look similar to this example:

```
<?xml version="1.0" encoding="utf-8"?>
<AdminStudioRules xmlns="http://adminstudio.com/AdminStudio.Rules.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation=
"http://adminstudio.com/AdminStudio.Rules.xsd AdminStudio.Rules.xsd">
  <Rule title="My Rule Title" description="My Rule Description"
author="Author" provider="Any">
    <Fingerprint name="Always Satisfied" description=""> </Fingerprint>
    <Action description="My First Action">
        <Message id="12000" text="I will automate my knowledge" severity="Warning"/>
    </Action>
  </Rule>
</AdminStudioRules>
```

6. Build any virtual package and you will see your Warning message.

Custom Rules Reference

This section includes descriptions of the Rule elements that you can use to create a custom Rule for use in virtual package testing and conversion using Windows Installer Converter.

- [Rule Element](#)
- [Fingerprint Element](#)
- [Registry Element](#)
- [File Element](#)
- [Shortcut Element](#)
- [Action Element](#)
- [Message Element](#)
- [Exclude Element](#)
- [FullyIsolate Element](#)
- [PunchThrough Element](#)
- [Create Element](#)

Rule Element

A Rule element is the basic definition for a rule. It is a child element of the AdminStudioRules element. If you wanted to define multiple rules, you would simply create multiple Rule elements. Here is an example of a Rule element:

```
<Rule title="MyRuleTitle" description="MyRuleDescription" author="Author" provider="Any">
</Rule>
```

The Rule Element has both attributes and child elements.

Attributes

The Rule element has the following attributes:

Table 9-12 • Rule Element / Attributes

Attribute	Description
Title	Name of Rule.
Description	Information that explains the purpose of the Rule.
Author	Person who authored the Rule.
Provider	Identifies the type of virtual package this Rule applies to: <ul style="list-style-type: none"> • Any—Rule will run for any provider type. • AppV—Rule will only run for App-V packages. • ThinApp—Rule will only run for ThinApp packages. • Citrix—Rule will only run for Citrix packages.

Child Elements

The Rule element has the following child elements:

Table 9-13 • Rule Element / Child Elements

Element Name	Description
Fingerprint	A Fingerprint is the condition associated with the Rule. If the condition is satisfied, an Action will be taken. <ul style="list-style-type: none"> • A Rule must have at least one Fingerprint child element. • If a rule has more than one Fingerprint child element they are logically OR'd. Therefore, if any one of the Fingerprints is satisfied, then the Action will be executed. For more information, see Fingerprint Element .
Action	An Action element is a container for child elements that perform an action if the Fingerprint condition is satisfied. A Rule must have one and only one Action element. For more information, see Action Element .

Fingerprint Element

A Fingerprint is the condition associated with the Rule. The following is a sample Fingerprint element:

```
<Fingerprint name="Adobe Distiller AdobePDFSettings" description="">
  <Registry hive="HKEY_CURRENT_USER" key="Software\Adobe\Acrobat Distiller\AdobePDFSettings\*" />
</Fingerprint>
```

If the condition is satisfied, an Action will be taken.

- A Rule must have at least one Fingerprint child element.
- If a rule has more than one Fingerprint child element they are logically OR'd. Therefore, if any one of the Fingerprints is satisfied, then the Action will be executed.

Note the following about the Fingerprint element:

- **Logical AND**—When evaluating whether a Fingerprint is satisfied, all of the Registry and File child elements are logically AND'd. Thus every child element must exist in the package in order for the Fingerprint to be satisfied.
- **Empty Fingerprint**—A Fingerprint with no child elements will always be satisfied. This is referred to as an empty Fingerprint.
- **File/Registry attributes are used as matching criteria**—When determining whether a Fingerprint is satisfied, only the attribute specified in the File or Registry child element will be used as matching criteria. For example, if you define a Registry child element with only the type="REG_SZ" attribute, then that item will be satisfied if there are any registry entries in the package with that type string. Similarly, if you define a File child element with only the name="*.cpl" attribute, then that item will be satisfied if there are any files with a .cpl extension in your entire package.
- **Default attribute values are not read**—A Fingerprint element will only try to match the attribute values you provide. Defaults are not read for Fingerprints.

Attributes

The Fingerprint element has the following attributes:

Table 9-14 • Fingerprint Element / Attributes

Attribute	Description
Name	Name to identify the Fingerprint element.
Description	Description of the purpose of the Fingerprint.

Child Elements

The Fingerprint element has the following child elements:

Table 9-15 • Fingerprint Element / Child Elements

Element Name	Description
File	The File element is used when searching for files. A Fingerprint may have zero, one or many File child elements. For more information, see File Element .
Registry	The Registry element is used when searching for registry entries. A Fingerprint may have zero, one or many Registry child elements. For more information, see Registry Element .

Registry Element

The Registry element is used when searching for Registry entries. The following is a sample Registry element:

```
<Registry hive="HKEY_LOCAL_MACHINE" key="System\CurrentControlSet\Services\Tcpip\*" />
```

The Registry element has the following attributes:

Table 9-16 • Registry Element / Attributes


Attribute	Description
ID	<p>The ID attribute is always used for binding. Binding allows you to reference an object from another object. You can bind together File, Registry, Shortcut, and Message text. For example, if you wanted to create a shortcut to a file, you would use the ID attribute to identify the file, and then obtain the shortcut for that file (the full path and file name).</p> <p>The ID attribute is not required.</p>  <p>Note • You should not use an ID attribute unless you are going to bind to it because it adds overhead to the conversion process.</p>
hive	<p>The hive attribute is the C++ constant name of the registry hive. Most likely, the hive would be one of the following:</p> <p>HKEY_CLASSES_ROOT HKEY_CURRENT_CONFIG HKEY_CURRENT_USER HKEY_LOCAL_MACHINE HKEY_PERFORMANCE_DATA HKEY_PERFORMANCE_NLSTEXT HKEY_PERFORMANCE_TEXT HKEY_USERS</p>
key	<p>The key attribute is the Registry key name, delimited by back slashes. For example:</p> <ul style="list-style-type: none"> • If you are using keys in a Fingerprint, you can end your key attribute with a * (back slash asterisk) and that will match all subkeys. • If you enter <code>software\Microsoft*</code>, that Fingerprint will match anything that is a subdirectory of software\Microsoft. <p>The key field is a valid binding target. See Rule Data Binding.</p>
valueName	<p>The valueName attribute is the name of the registry value.</p> <p>The valueName field is a valid binding target. See Rule Data Binding.</p>
valueData	<p>The value Data attribute is a string representation of your value data. If you are creating a Fingerprint, this is the information that would be returned by RegQueryValueEx, which retrieves the type and data for the specified value name associated with an open registry key.</p> <p>The valueData field is a valid binding target. See Rule Data Binding.</p>

Table 9-16 • Registry Element / Attributes

Attribute	Description
valueType	The valueType attribute specifies a string that represents the data type for a registry entry. If you do not specify a valueType, the default valueType is REG_SZ.

File Element

The File element is used when searching for files. The following are sample File elements:

```
<File name="vmware-vmx.exe"/>
<File id="cp1File" name="*.cp1"/>
```

The File element has the following attributes:

Table 9-17 • File Element / Attributes


Attribute	Description
id	<p>The id attribute is always used for binding. Binding allows you to reference an object from another object. You can bind together File, Registry, Shortcut, and Message text. For example, if you wanted to create a shortcut to a file, you would use the id attribute to identify the file, and then obtain the shortcut for that file (the full path and file name).</p> <p>The id attribute is not required.</p>  <p>Note • You should not use an id attribute unless you are going to bind to it because it adds overhead to the conversion process.</p>
name	The name attribute is the name of the file with its extension. However, you can also use file wild cards when specifying the name attribute. For example, if MyFile.* was entered, all of the files that start with MyFile would be found.
directory	<p>The directory attribute identifies the name of the directory that contains the file.</p> <ul style="list-style-type: none"> You can use variables in a directory attribute, which are similar to Windows Installer folder variables, except instead of using square brackets, you use percent signs (%). For example, if you wanted to check for a file in the windows folder, you would enter %Windows Folder%. Also, similar to registry key, if you end a directory attribute with *, all subdirectories will be searched. By default, only the specified directory will be searched. <p>This field is a valid binding target. See Rule Data Binding.</p>
minVersion	<p>The minVersion attribute identifies the minimum file version. It is inclusive.</p> <ul style="list-style-type: none"> If the minVersion is 1.0, it will match files that have a minimum version of 1.0.

Table 9-17 • File Element / Attributes

Attribute	Description
maxVersion	<p>The maxVersion attribute identifies the maximum file version. It is inclusive.</p> <ul style="list-style-type: none"> • If the maxVersion is 4.0, it will match files that have a maximum version of 4.0. • If you specify a minVersion but not a maxVersion, any file that has a version higher than the minVersion will be found. • If you don't specify any minVersion or maxVersion, there will not be a match on these attributes.

Shortcut Element

The Shortcut element is used to create shortcuts. The following is an example Shortcut element:

```
<Shortcut name="{BIND:cp1File FileName}" target="%SystemFolder%\rundll32.exe"
arguments="shell32.dll,Control_RunDLL "{BIND:cp1File FullPath}"" workingDir="" />
```

The Shortcut element has the following attributes:

Table 9-18 • Shortcut Element / Attributes

Attribute	Description
name	The name attribute identifies the file name of the file with its extension. However, you can use file wild cards. For example, MyFile.* will find all files that start with MyFile.
target	The target element identifies the location of the shortcut that is being created.
arguments	The arguments attribute contains the command line parameters for the Shortcut element.
workingDir	The workingDir attribute identifies the default current directory for the application.

Action Element

An Action element is essentially just a container for its child elements. The child elements actually perform the action. An Action element with no child elements will do nothing. The following is a sample Action element:

```
<Action description="Exclude AdobePDFSettings">
  <Message id="11006" text="Excluding the registry key Software\Adobe\Acrobat Distiller\
  AdobePDFSettings. Adobe Distiller will recreate these settings on first use." severity="Fix"/>
  <Exclude>
    <Registry hive="HKEY_CURRENT_USER" key="Software\Adobe\Acrobat Distiller\AdobePDFSettings\*" />
  </Exclude>
</Action>
```

Attributes

The Action element has the following attributes:

Table 9-19 • Action Element / Attributes

Attribute	Description
description	Allows you to annotate your action. It is not used by the Rules Engine.

Child Elements

The Action element has the following Child Elements:

Table 9-20 • Action Element / Child Elements

Element Name	Description
Message	The Message element specifies the message that is displayed in the Build Output window. For more information, see Message Element .
Exclude	You can use the Exclude element to exclude files or registry entries from the virtual package. For more information, see Exclude Element .
FullyIsolate	If you use the FullyIsolate element, the specified resource will not be able to interact with the underlying system. If you fully isolate a directory, the virtual package will not be able to see anything in that isolated directory. For more information, see FullyIsolate Element .
PunchThrough	If you use the PunchThrough element, any write operation will be written to the physical machine and not into the virtual container. If you PunchThrough a directory, any files that you write to that directory will go to the machine instead of to the virtual package sandbox. For more information, see PunchThrough Element .
Create	Use the Create element to add files, registry entries, and shortcuts to the virtual package. For more information, see Create Element .

Message Element


The Message element will send a message to the build engine. These messages are registered as build output and, in addition, will also be registered as Pre-Conversion Evaluation output. The following is a sample Message element:

```
<Message id="12001" text="Excluding TCPIP registry entries from the Citrix profile" severity="Fix"/>
```

Using the Message element is the *only* way an Action can communicate with the user. All of the other Action child elements are completely silent.

The Message element has the following attributes:

Table 9-21 • Message Element / Attributes

Attribute	Description
id	<p>The ID attribute identifies the message, and it is required.</p>  <p>Note • The message IDs reserved for user rules are 12000 and above. Therefore, when writing your own custom rules, assign to Message IDs a value greater than 12000.</p>
text	<p>The text attribute contains the message text that is displayed to the user. This field is a valid binding target. See Rule Data Binding.</p>
severity	<p>The severity attribute identifies the message's degree of importance:</p> <ul style="list-style-type: none"> • Fatal • Error • Warning • Fix • Information

Exclude Element

The Exclude element excludes items from the package. The Exclude action will match File and Registry items using the same rules as Fingerprint matching. The following is a sample Exclude element:

```
<Exclude>
  <Registry hive="HKEY_LOCAL_MACHINE" key="System\CurrentControlSet\Services\Tcpip\*" />
</Exclude>
```

The Exclude Element has the following child elements:

Table 9-22 • Exclude Element / Child Elements

Element Name	Description
File	<p>The File element is used when searching for files. For more information, see File Element.</p>
Registry	<p>The Registry element is used when searching for registry entries. For more information, see Registry Element.</p>

FullyIsolate Element

Full isolation means that the virtual application will not be able to see the specified resource if it exists on the target machine. The following is a sample FullyIsolate element:

```
<FullyIsolate>
  <Registry hive="HKEY_LOCAL_MACHINE" key="System\CurrentControlSet\Services\Tcpip\*" />
</FullyIsolate>
```

The FullyIsolate element sets the Full Isolation virtualization attribute on the specified items from the package. The FullyIsolate action will match File and Registry items using the same rules as Fingerprint matching. This action is subject to the limitations of the virtualization technology; for instance you can only set isolation objects at the registry key level and not at the registry value level.

The FullyIsolate element has the following child elements:

Table 9-23 • FullyIsolate Element / Child Elements

Element Name	Description
File	The File element is used when searching for files. For more information, see File Element .
Registry	The Registry element is used when searching for registry entries. For more information, see Registry Element .

PunchThrough Element

Punch Through means that the virtual application will be able to write to that area of the target machine. The following is a sample PunchThrough element:

```
<PunchThrough>  
  <Registry hive="HKEY_LOCAL_MACHINE" key="System\CurrentControlSet\Services\Tcpip\*" />  
</PunchThrough>
```

The PunchThrough element sets the Punch Through virtualization attribute on the specified items from the package. The PunchThrough action will match File and Registry items using the same rules as Fingerprint matching. This action is subject to the limitations of the virtualization technology; for instance you can only set isolation objects at the registry key level and not at the registry value level.

The PunchThrough element has the following child elements:

Table 9-24 • PunchThrough Element / Child Elements

Element Name	Description
File	The File element is used when searching for files. For more information, see File Element .
Registry	The Registry element is used when searching for registry entries. For more information, see Registry Element .

Create Element

The **Create** element creates the items defined by the child elements if they do not already exist. There is a minimum level of information required to create an item. The following is a sample Create element:

```
<Create>  
  <Shortcut name="{BIND:cp1File FileName}" target="%SystemFolder%\rundll32.exe"  
    arguments="shell32.dll,Control_RunDLL &quot;{BIND:cp1File FullPath}&quot;;" workingDir="" />  
</Create>
```

The Create element has the following Child Elements:

Table 9-25 • Create Element / Child Elements

Element Name	Description
File	The File element is used when searching for files. Only directories can be created. For more information, see File Element .
Registry	The Registry element is used when searching for registry entries. It minimally requires the hive and key attributes. For more information, see Registry Element .
Shortcut	The Shortcut element is used to create shortcuts. It minimally requires the name and target attributes. For more information, see Shortcut Element .

Rule Data Binding

In Rule data binding, you have a binding source and a binding target. The binding source is the item that you put an ID on. For example, if you set the ID of a File element to F1, you can then use that File element as a binding source. The binding target is the item that references the source.

Binding Source

Only a Registry entry or a File can be a binding source. The only attributes on a File element that you can bind to are filename, folder, and fullpath.

Binding Source for a File

The following is an example of a binding source for a file:

```
<File id="F1" ... />
  {Bind:F1 filename} = Notepad.exe
  {Bind:F1 folder}   = C:\Windows
  {Bind:F1 fullpath} = C:\Windows\Notepad.exe
```

For example, if you have the File attribute and it finds the file C:\Windows\Notepad.exe, the Bind command for filename would return Notepad.exe. The Bind command for folder would return C:\Windows.

Binding Source for a Registry Entry

The following is an example of a binding source for a registry entry:

```
<Registry id="R1" ... />
  {Bind:R1 hive}      = HKEY_CURRENT_USER
  {Bind:R1 key}       = Environment
  {Bind:R1 valueName} = TEMP
  {Bind:R1 valueType} = REG_SZ
  {Bind:R1 valueData} = %USERPROFILE%\LocalSettings\Temp
```

For example, suppose you have the Registry attribute and it finds the following registry entry:

```
[HKEY_CURRENT_USER\Environment]  
TEMP="%USERPROFILE%\LocalSettings\Temp"
```

The Bind command for hive would return HKEY_CURRENT_USER. The Bind command for key would return Environment.

Binding Target

The binding target is the item that references the binding source. The following is an example of a binding target:

```
<Registry hive="HKEY_CURRENT_USER" key="Software\Test" valueName="{Bind:F1 fullPath}"/>
```

Example

The following is an example using Rule Binding that finds a file and then creates a shortcut to it:

Table 9-26 • Example Rule Using Rule Binding

```
<Rule title="Control Panel Applet - ThinApp" description="Create a default shortcut for  
control panel applets" author="AdminStudio.Default" provider="ThinApp">  
  <Fingerprint name="Detect Control Panel Applet" description="">  
    <File id="cp1File" name="*.cpl" minVersion="0.1" />  
  </Fingerprint>  
  <Action description="Create Control Panel Applet Shortcut">  
    <Message id="11004" text="Generating shortcut for the Control Panel Applet located  
    at '{BIND:cp1File FullPath}'" severity="Fix" />  
    <Create>  
      <Shortcut name="{BIND:cp1File FileName}" target="%SystemFolder%\rundll32.exe"  
      arguments="shell32.dll,Control_RunDLL \"{BIND:cp1File FullPath}\"" workingDir="" />  
    </Create>  
  </Action>  
</Rule>
```

Testing App-V Applications Using the App-V Application Launcher

The App-V Application Launcher is a convenient testing tool that makes it possible for you to reliably and accurately test your App-V applications on your local machine or any other system that has the App-V client installed before moving it to the App-V server.

App-V Application Launcher Utility Location

When an App-V application is built, the App-V Application Launcher utility (AppVLauncher.exe) is placed in the same folder as the App-V application (the same directory that contains the .sft and .osd files).

Chapter 9: Converting Windows Installer Packages to Virtual Applications

Testing App-V Applications Using the App-V Application Launcher



Figure 9-10: App-V Application Launcher in the App-V Application Output Directory

Opening the App-V Application Launcher Utility

To open the App-V Application Launcher, double-click on the App-V Application Launcher utility (AppVLauncher.exe). The App-V Application Launcher will attempt to launch that application. If there are multiple shortcuts (.osd files) in this App-V application, the **Launch App-V Application** dialog box opens, where you are prompted to select the shortcut you want to launch from a list of all of the shortcuts.

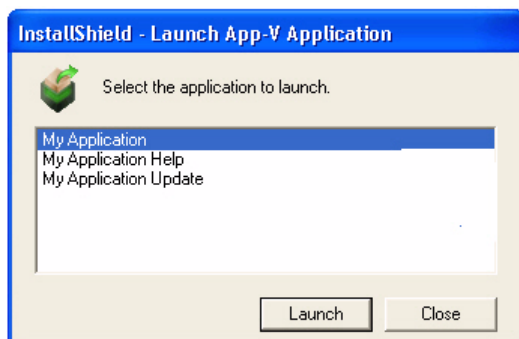
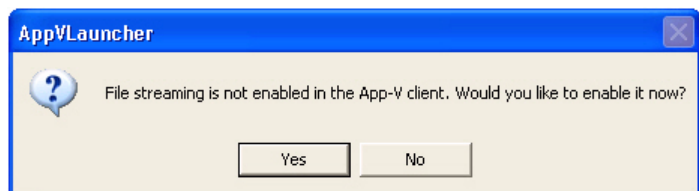


Figure 9-11: Launch App-V Application Dialog Box

Requirements for Using the App-V Application Launcher

There are two requirements for using the App-V Application Launcher:

- **Microsoft Virtualization client must be installed**—In order to test an App-V application using the App-V Application Launcher, you must have the Microsoft Virtualization Client installed on the machine.
- **File streaming must be enabled**—If you attempt to launch the App-V Application Launcher without having App-V file streaming enabled, an error message will open prompting you to enable the file streaming option. Click **Yes** to enable it.



Controlling Whether to Include the App-V Application Launcher in Output

If you do not want to include the App-V Application Launcher with each App-V application that you build, clear the **Create utility to test launch App-V applications as part of building App-V applications** check box on the **Virtual Packages** tab of the AdminStudio **Options** dialog box.

Running the Windows Installer Converter from the Command Line

The executable that launches the Windows Installer Converter is:

```
[AdminStudio Installation Directory]\Repackager\msi2vp.exe
```

The syntax for running the Windows Installer Converter on the command line is:

- **Microsoft App-V application**—To build a Microsoft App-V application, enter:
`msi2vp.exe -s -i <path to .msi file> -t <transform1|transform2|...> -v appv`
- **ThinApp application**—To build a ThinApp application, enter:
`msi2vp.exe -s -i <path to .msi file> -t <transform1|transform2|...> -v thinapp`
- **Citrix profile**—To build a Citrix profile, enter:
`msi2vp.exe -s -i <path to .msi file> -t <transform1|transform2|...> -v citrix`
- **Both**—To build all three virtual package formats, enter:
`msi2vp.exe -s -i <path to .msi file> -t <transform1|transform2|...> -v appv|thinapp|citrix`

The command line parameters are described in the following table.

Table 9-27 • Windows Installer Converter Command Line Options

Parameter	Description
-s	Use the -s parameter to indicate you want to perform this conversion silently, with no user interface and no user input required.

Table 9-27 • Windows Installer Converter Command Line Options

Parameter	Description
-i <path to .msi file> -i <path to directory root containing .msi files>	<p>Use the -i parameter followed by a directory location to specify either:</p> <ul style="list-style-type: none"> • Single Application option—The Windows Installer package you want to convert to a virtual application. • Batch Convert option—The directory that contains multiple Windows Installer packages that you want to convert to individual virtual applications. It is recommended that each of these Windows Installer packages be in a separate subdirectory of the selected directory.  <p>Note • The Batch Convert option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.</p>
-m "app1.msi app2.msi app3.msi"	<p>Virtual Suite option—Use the -m parameter followed by the path to the packages that you want to include in the Virtual Suite separated by pipes. Here are some examples:</p> <p>-m "c:\packages\app1.msi c:\packages\app2.msi"</p>
-t <transform1 transform2 ...>	<p>Use the -t parameter followed by the name of a transform file to include it in the virtual application. To enter more than one transform file, separate them by pipes.</p>  <p>Note • It is preferable that all of the transforms that you include in a virtual package be located in the same folder as the Windows Installer .msi package so that they can be accessed when the virtual package is built.</p>
-v appv	Enter -v appv to generate a Microsoft App-V application.
-v thinapp	Enter -v thinapp to generate a ThinApp application.
-v citrix	Enter -v citrix to generate a Citrix profile.
-v appv thinapp citrix	Enter -v appv thinapp citrix to generate packages in all three virtual package formats.

Capturing Virtualization Context

Sometimes it is necessary to repackage a Windows Installer package before you can successfully virtualize it (as described in [Virtualization Conversion Error Messages](#)).

When some Windows Installer packages are repackaged, some of their data (such as files or registry entries) are excluded according to the normal Repackager exclusion settings. For example, files destined for the `\Windows\Installer` folder are typically excluded. However, this type of information is occasionally necessary in order to successfully convert a Windows Installer package to a virtual package.

To address this issue, when Repackager builds a Windows Installer package, it now produces two `.msi` files: `packagename.msi` and `packagename.context.msi`.

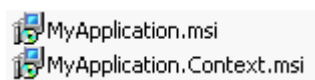


Figure 9-12: Repackaged Output: application.msi and application.context.msi



Important • If you are not converting a package to a virtual package, you can ignore its `.context.msi` file.



Note • Context data is not displayed in the Repackager interface when viewing captured Files/Registry details.

How is the Context Data Configured

Context data is configured in a settings file in the AdminStudio Shared folder called `isrepackager.context.ini`. It is identical in syntax to the familiar `isrepackager.ini` file which is used for exclusion settings. Data that matches the settings in the `context.ini` file is captured—not into the main application `.msi` file, but rather into a separate `context.msi` file. When creating a virtual package, Repackager combines the data in both the main `.msi` file and the `.context.msi` file to produce the final virtual package.

Windows Installer Converter Reference

This section describes each of the panels in the Windows Installer Converter, and contains a list of the error messages that could occur when building a virtual package.

- [Windows Installer Converter](#)
- [Virtualization Conversion Error Messages](#)

Windows Installer Converter

You can use the Windows Installer Converter to convert Windows Installer `.msi` packages to Microsoft App-V applications, ThinApp applications, or Citrix profiles. You have three conversion options:

- **Single Application**—Convert one Windows Installer package into a virtual package.
- **Virtual Suite**—Convert multiple Windows Installer packages, perhaps a group of applications that need to work together, into one virtual package.
- **Batch Convert**—Convert a directory full of Windows Installer packages into individual virtual packages.



Edition • The **Batch Convert** option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.

These virtual applications can be distributed throughout your organization. Users will be able to run the virtual application in an isolation environment that does not conflict with their operating system or existing applications.

You can also use Windows Installer Converter's Pre-Conversion Evaluation feature to test applications for virtualization readiness prior to proceeding with conversion.



Note • You can also convert a Windows Installer package to a virtual application using InstallShield Editor's Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant. Using the InstallShield Editor Assistants, you can configure a virtual application's files, folders, shortcuts, registry settings, isolation options, and build options. See [Getting Started With Application Virtualization](#) and the InstallShield Help Library for information.

The Windows Installer Converter includes the following panels:

- [Welcome Panel](#)
- [Select Conversion Target Panel](#)
- [Conversion Mode Panel](#)
- [Select Windows Installer Package](#)
- [Select a Directory Panel](#)
- [Select a Suite of Packages Panel](#)
- [Test Package\(s\) Panel](#)
- [Edit Windows Installer Package Panel](#)
- [Package Conversion Panel](#)

Welcome Panel

You can use the Windows Installer Converter to convert a Windows Installer .msi package to a Microsoft App-V application, a ThinApp application, or a Citrix profile for Citrix XenApp.

- You could then distribute the virtual application throughout your organization.
- Users will be able to run the application in an isolation environment that does not conflict with their operating system or existing applications.
- You can also use the Pre-Conversion Evaluation feature of Windows Installer Converter to test packages for virtualization readiness prior to conversion.

On the **Welcome** panel, click **Next** to continue.

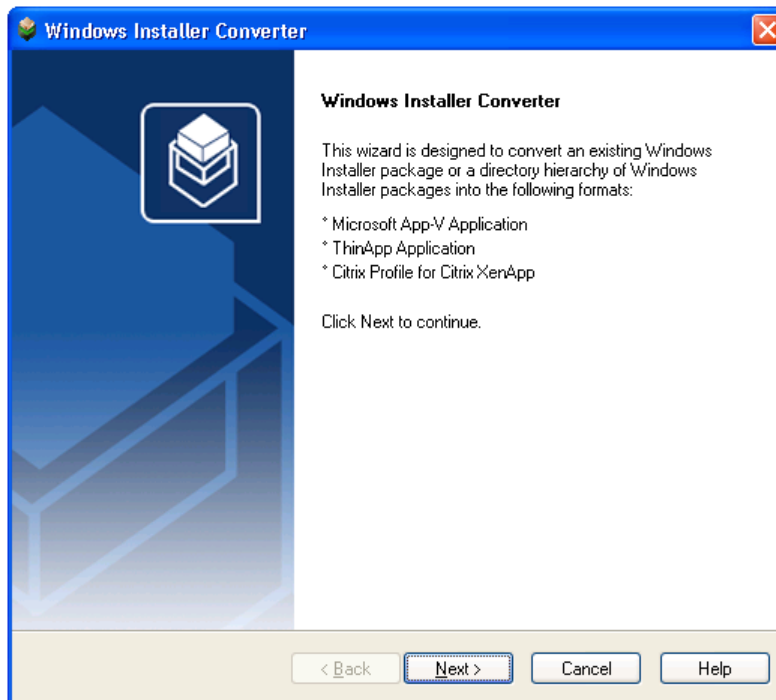


Figure 9-13: Windows Installer Converter / Welcome Panel



Note • You can also convert a Windows Installer package to a virtual application using InstallShield Editor's Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant. Using the InstallShield Editor Assistants, you can configure a virtual application's files, folders, shortcuts, registry settings, isolation options, and build options. See [Getting Started With Application Virtualization](#) and the InstallShield Help Library for information.

Select Conversion Target Panel

On the **Select Conversion Target** panel, select the virtual application formats that you would like to build: **Microsoft App-V Application**, **ThinApp Application**, and/or **Citrix Profile**.

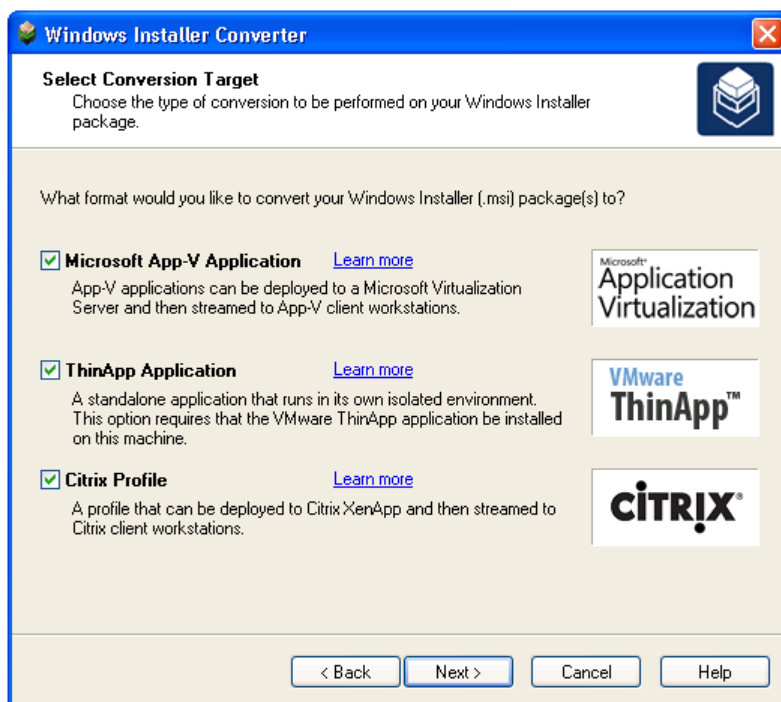


Figure 9-14: Windows Installer Converter / Select Conversion Target Panel

For more information on creating and deploying virtual applications, click the [Learn more](#) links on this panel.



Important • To create a *ThinApp* application, you must have *ThinApp* installed on the current machine. If you do not, the **ThinApp Application** option will be disabled. For more information, see the [VMware Web site](#).

Conversion Mode Panel

On the **Conversion Mode** panel, you identify the package input/output mode and whether you want to test those packages prior to conversion.

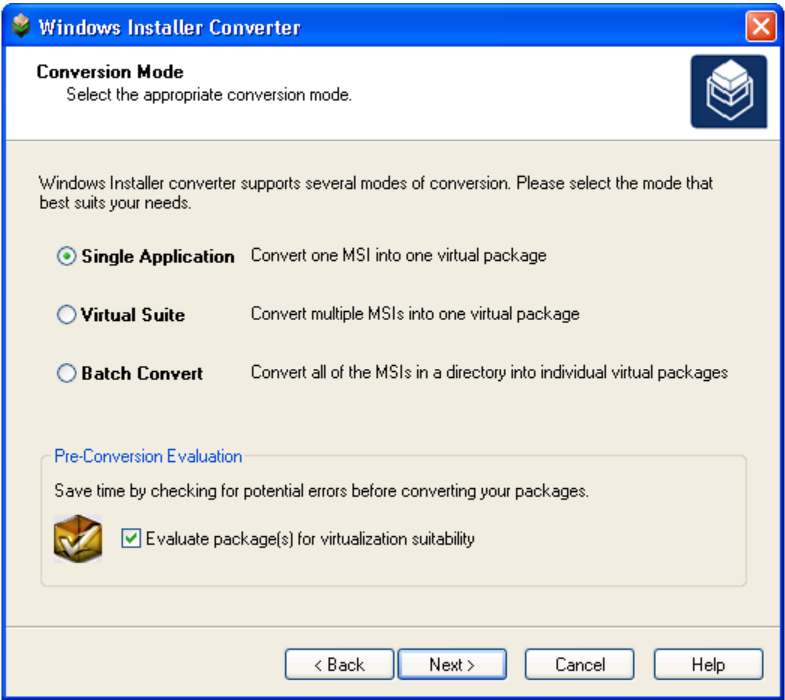



Figure 9-15: Windows Installer Converter / Conversion Mode Panel

This panel includes the following options:

Table 9-28 • Windows Installer Converter / Conversion Mode Panel

Option	Description
Single Application	Select this option if you want to convert one Windows Installer package into a virtual package.

Table 9-28 • Windows Installer Converter / Conversion Mode Panel

Option	Description
Virtual Suite	<p>Select this option if you want to convert multiple Windows Installer packages, perhaps a group of applications that need to work together, into one virtual package.</p> <p>When creating a virtual suite of packages, the following occurs:</p> <ul style="list-style-type: none"> • Microsoft App-V—When creating a Microsoft App-V application, Windows Installer Converter converts the installation data from the original Windows Installer package into an .sft file that is compatible with Microsoft App-V version 4.5. The installation shortcuts are converted into .osd files, which provide access to programs within the .sft file. • ThinApp—When creating a virtual suite of packages using ThinApp, Windows Installer Converter creates a program shortcut for every application in the suite and all shortcuts are presented to end users. • Citrix—When creating a virtual suite of packages using Citrix XenApp, Windows Installer Converter also creates a program shortcut for every application in the suite. The .profile file contains the shortcut information, and the .cab file contains all of the files necessary to run the applications in the suite. However, when you distribute a virtual package using Citrix XenApp, you can only select one shortcut. If you want this virtual suite to have more than one shortcut, you will need to distribute it more than once.
Batch Convert	<p>Select this option if you want to convert a directory full of Windows Installer packages into individual virtual packages.</p>  <p>Edition • The Batch Convert option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.</p>
Evaluate package(s) for virtualization suitability	<p>Select this option to evaluate the package(s) prior to conversion. After viewing the results, you can choose the packages you want to convert.</p>

Select Windows Installer Package

On the **Select Windows Installer Package** panel, which opens when you select **Single Application** on the **Conversion Mode** panel, click **Browse** and select the Windows Installer .msi package that you want to convert to a virtual application.

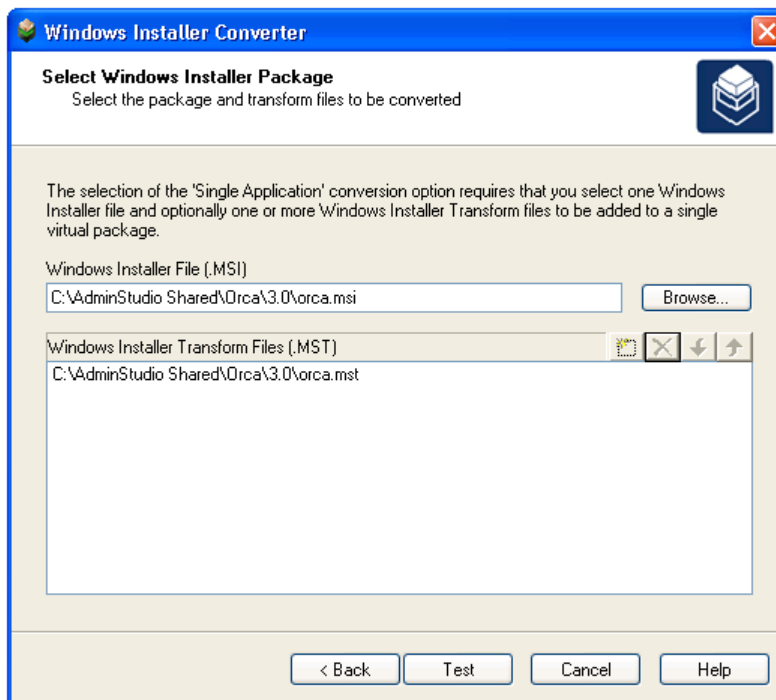


Figure 9-16: Windows Installer Converter / Select Windows Installer Package Panel

All of the transform files that are in the same folder as the selected Windows Installer package are automatically listed in the **Windows Installer Transform Files (.MST)** list.



Note • It is preferable that all of the transforms that you include in a virtual package be located in the same folder as the Windows Installer .msi package so that they can be accessed when the virtual package is built. Only those transform files that are in this folder will be listed on the **Transform Files (.MST)** list on this panel.

Select those transform files that you would like to include in the virtual package.

- The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up (↑) and Move Down (↓) buttons.
- Use the Delete button (✕) to delete a transform from the list.

Select a Directory Panel



Edition • The **Batch Convert** option of Windows Installer Converter is only available when purchased on the AdminStudio desktop licensing model.

On the **Select a Directory** panel, which opens when you select **Batch Convert** on the **Conversion Mode** panel, click **Browse** and select a single directory whose contents include at least one Windows Installer file. Any Windows Installer file(s) will be converted into individual virtual packages.

After you have made your selection, click **Convert** or **Test** to continue.

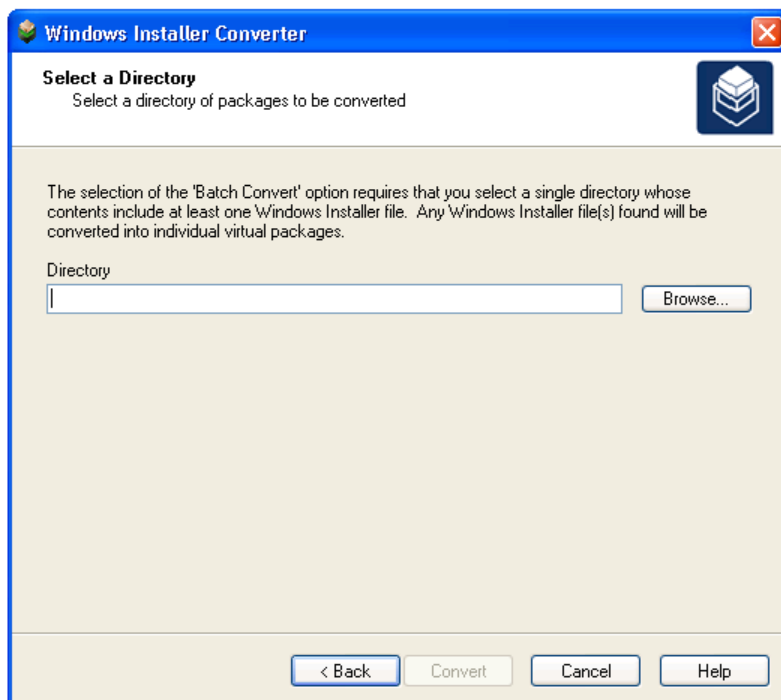


Figure 9-17: Windows Installer Converter / Select a Directory Panel

After you select the directory, the number of packages that Windows Installer Converter finds in that directory are listed.

Select a Suite of Packages Panel

When you select the **Virtual Suite** option on the **Conversion Mode** panel, the **Select a Suite of Packages** panel opens and prompts you to select:

- **Primary Windows Installer File**—Package in the virtual suite that you want the end users to primarily interact with.
- **Additional Suite Windows Installer Files**—Packages that are needed directly by the primary package and are used only by that package—such as driver files or client components. These packages are not part of the common desktop environment in the Enterprise.

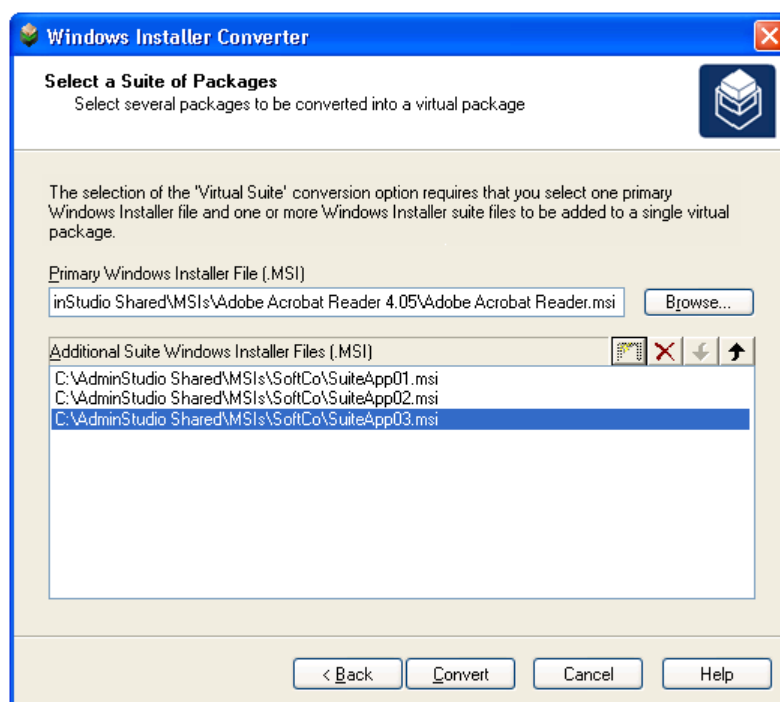





Figure 9-18: Windows Installer Converter / Select a Suite of Packages Panel



Note • The creation of application shortcuts varies depending upon which virtual application format you are using:

- **App-V**—When creating a Microsoft App-V application, Windows Installer Converter converts the installation data from the original Windows Installer package into an .sft file that is compatible with Microsoft App-V version 4.5. The installation shortcuts are converted into .osd files, which provide access to programs within the .sft file.
- **ThinApp**—When creating a virtual suite of packages using ThinApp, Windows Installer Converter creates a program shortcut for every application in the suite and all shortcuts are presented to end users.
- **Citrix**—When creating a virtual suite of packages using Citrix XenApp, Windows Installer Converter also creates a program shortcut for every application in the suite. However, when you distribute a virtual package using XenApp, you can only select one shortcut. If you want this virtual suite to have more than one shortcut, you will need to distribute it more than once.

When adding packages to this list, note the following:

- Each time you want to add an **Additional Suite Windows Installer Package**, click the **New** button () and browse to the package you want to add.
- Use the Up and Down arrows () to set the order in which the packages are converted. If there is a conflict between two packages, the package nearest the top of the list takes precedence.
- Use the Delete button () to delete a package from the list.

Click **Convert** when ready to begin conversion.

Test Package(s) Panel

If you select the **Evaluate package(s) for virtualization suitability** option on the **Conversion Mode** panel of the Windows Installer Converter, the **Test Package(s)** panel will open and the selected package(s) will be tested for virtualization compatibility before proceeding with conversion.

When testing is complete, the results for each package are displayed:

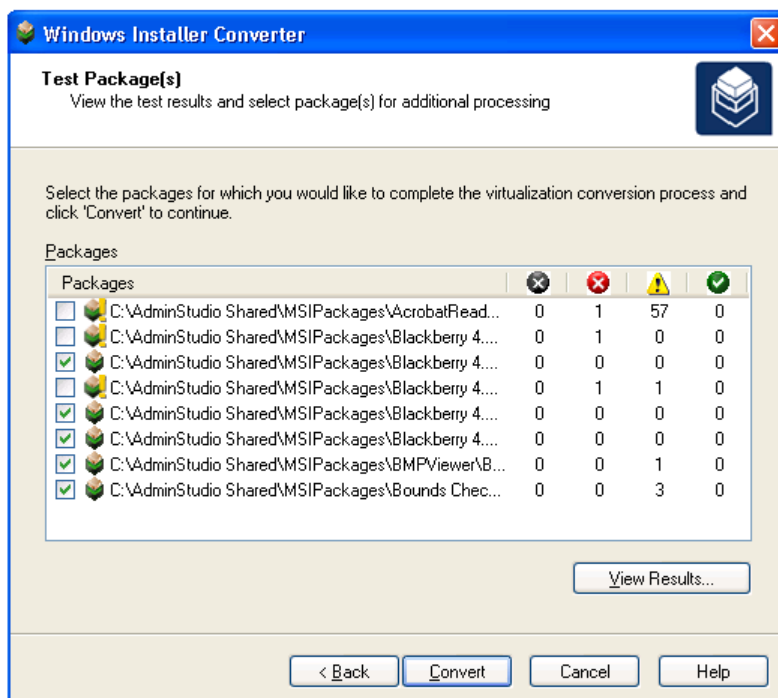


Figure 9-19: Windows Installer Converter / Test Package(s) Panel

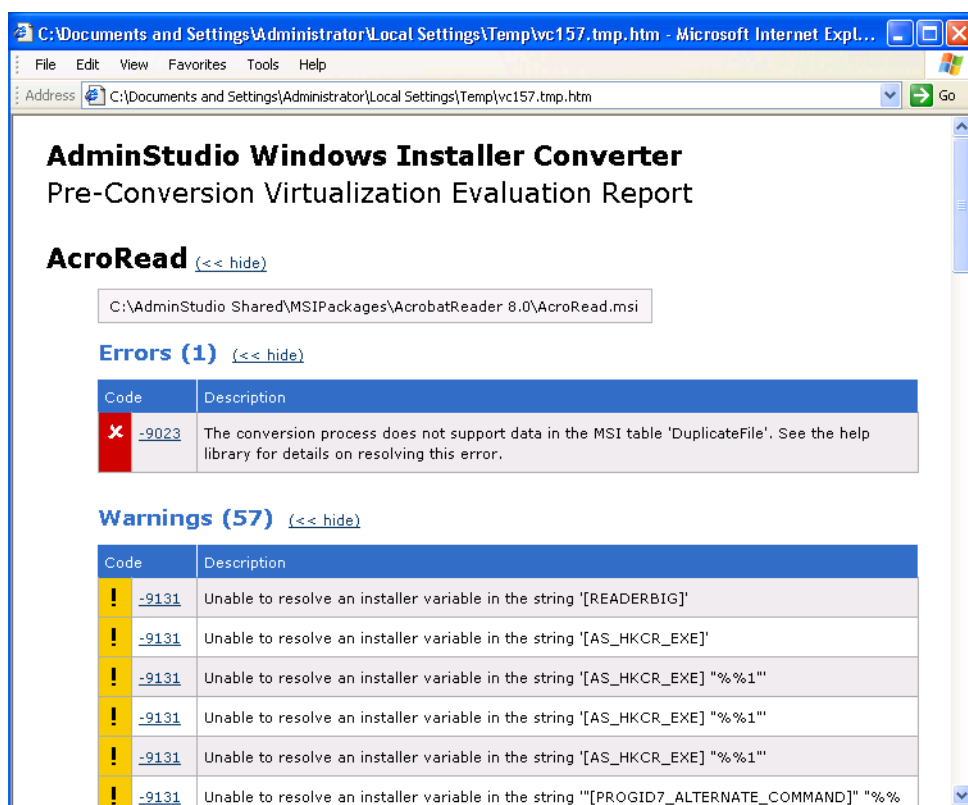
The number of errors and warnings generated are displayed under the appropriate icon:

Table 9-29 • Message Icons on the Test Package(s) Panel of Windows Installer Converter

Icon	Name	Description
	Fix	Windows Installer Converter discovered an issue and resolved it.
	Warning	The virtual package will probably run, but may not.
	Error	The virtual package probably won't run, but under some circumstances it may.
	Fatal Error	The virtual package will not run.

Those packages that did not have any errors or fatal errors are automatically selected and the **Convert** button is enabled.

To see an expanded view of the test results, click the **View Results** button. An HTML report is generated that lists all of the messages and provides links to a help topic for each message:



Because this report is in HTML format, it can be saved and reviewed by others. To save the HTML report, select **Save As** from the **File** menu.

After reviewing the test results you can decide which packages you want to convert. Those that are deemed suitable for conversion will be pre-selected. You can then decide to proceed with conversion of the selected packages or wait until a later date to perform conversion.

Select those packages that you want to convert and click **Convert**. Conversion begins.

When package conversion is complete, a **Conversion completed** message appears in the **Output** window and the path to the generated virtual package(s) is provided.

Edit Windows Installer Package Panel

If you are an advanced user who wants a higher degree of customization, you can use the InstallShield Editor, Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant to make customizations before building the virtual application.



Important • If you are using standalone Repackager or if you have not installed InstallShield Editor on this machine, the **Edit Windows Installer Package** panel will not be displayed.

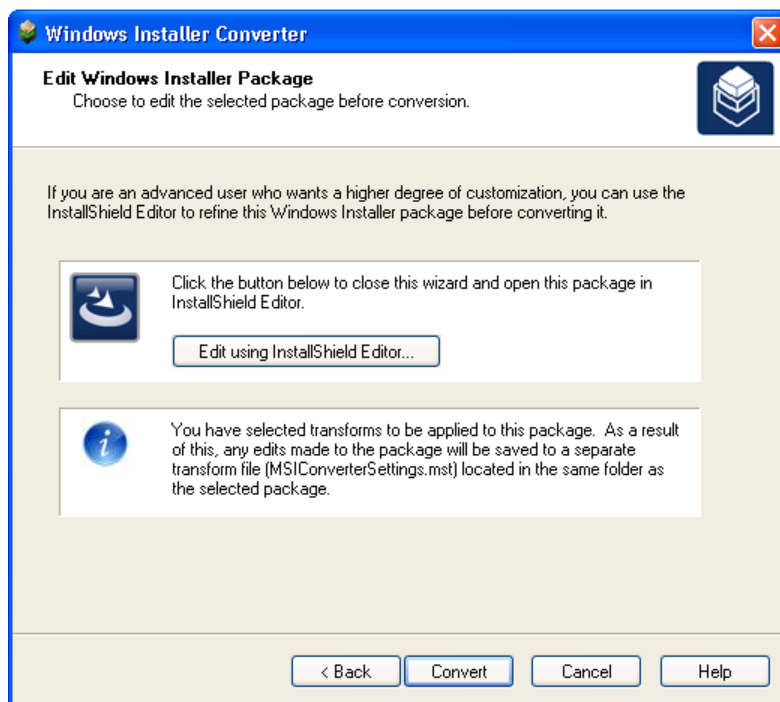


Figure 9-20: Windows Installer Converter / Edit Windows Installer Package Panel

To open the selected package in InstallShield Editor, click the **Edit Using InstallShield Editor** button. This wizard will close, and the package will open in InstallShield Editor in either Direct Edit Mode or in Direct MST Mode (if any transforms have been selected).

You would then click on the appropriate tab to use one of the Assistants to customize the package's isolation options and build the virtual application.

For more information, see [Creating Customized Virtual Applications](#).

Package Conversion Panel

The **Package Conversion** panel displays status messages in the **Output** window during the conversion process, and notifies you when package conversion is complete. When you click **Error List**, the **Error List** tab opens and displays the error and warning messages that were generated.

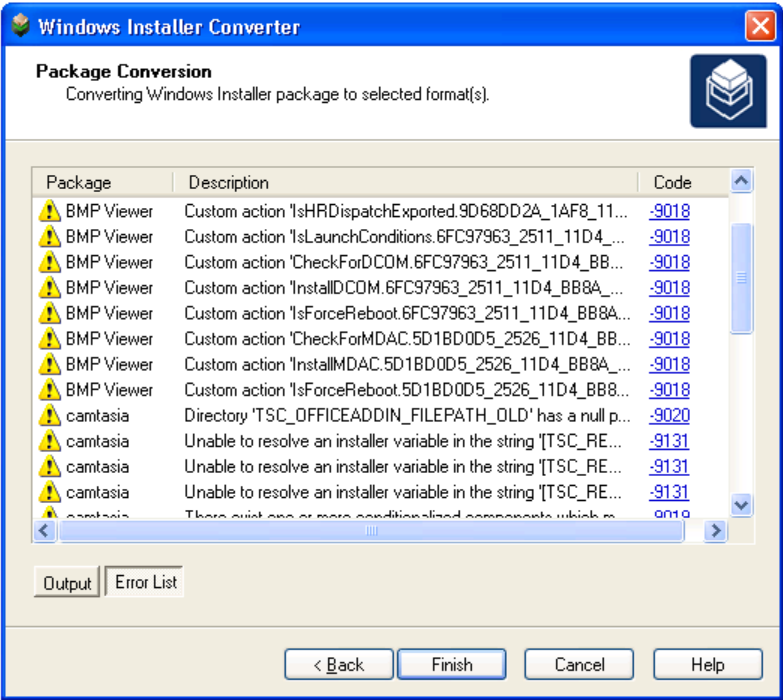


Figure 9-21: Error List Displayed on the Package Conversion Panel

The **Error List** tab of the Package Conversion panel includes the following information:

Table 9-30 • Error List Tab of Package Conversion Panel

Item	Description
Package	Name of package that generated the message.
Description	Text of error or warning message.
Code	Error or warning message number. You can click on the number to view a help topic that includes information on how to resolve that error, if possible. For a complete list of error and warning messages, see Virtualization Conversion Error Messages .

When package conversion is complete, click **Finish**.



Note • The Windows Installer Converter ignores services and user-defined custom actions. For more information, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

Virtualization Conversion Error Messages

When converting a Windows Installer package to a virtual application, error and warning messages are generated. Some of these messages are generic to package virtualization, and others are specific to the virtualization solution you are preparing packages for.

The error messages for Windows Installer Converter, App-V Assistant, VMware ThinApp Assistant, and Citrix Assistant

Error -9000: Unknown Exception

The following table documents this message:

Table 9-31 • Error -9000: Unknown Exception

Category	Description
Type:	Error
Message:	An unknown exception occurred.
Cause:	This is an unexpected internal error.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9001: Unknown COM

The following table documents this message:

Table 9-32 • Error -9001: Unknown COM

Category	Description
Type:	Error
Message:	Internal error.
Resolution:	Contact AdminStudio Technical Support.

Error -9002: Error Opening Package

The following table documents this message:

Table 9-33 • Error -9002: Error Opening Package

Category	Description
Type:	Error
Message:	An error occurred when opening the package.
Cause:	This is an unexpected internal error when reading the Windows Installer package.
Resolution:	<p>Check to make sure that the package is accessible to the user. If the error persists and the package is on a network share, copy the package locally (to avoid any network connection issues) and try again.</p> <p>If this does not solve the problem, perform these additional investigational steps and then contact AdminStudio Technical Support.</p>

Error -9003: Error Saving Package

The following table documents this message:

Table 9-34 • Error -9003: Error Saving Package

Category	Description
Type:	Error
Message:	An error occurred when saving the package.
Cause:	This is an unexpected internal error when trying to save the Citrix profile.
Resolution:	<p>Check to see if the user has proper access to the location the profile is being built to.</p> <p>If this does not solve the problem, perform these additional investigational steps and then contact AdminStudio Technical Support.</p>

Error -9004: Process Cancelled By User

The following table documents this message:

Table 9-35 • Error -9004: Process Cancelled By User

Category	Description
Type:	Error
Message:	Process cancelled by user.
Cause:	The user clicked the Cancel button to stop the build.
Resolution:	Restart the build process.

Error -9005: Error Creating Temporary Folder

The following table documents this message:

Table 9-36 • Error -9005: Error Creating Temporary Folder

Category	Description
Type:	Error
Message:	An error occurred while creating a temporary folder
Cause:	You encounter this error when the user does not have permission to write to C:\TMP, or the drive is out of disk space.
Resolution:	Obtain write access to C:\TMP, and free some disk space on the drive, and then rebuild the profile.

Error -9006: Error Decompressing Package

The following table documents this message:

Table 9-37 • Error -9006: Error Decompressing Package

Category	Description
Type:	Error
Message:	An error occurred while decompressing the package 'PackageName'.

Table 9-37 • Error -9006: Error Decompressing Package

Category	Description
Cause:	You encounter this error when the package is a compressed Windows Installer package (.msi) and errors were generated when AdminStudio attempted to perform an administrative installation to extract the files.
Resolution:	<p>When this error occurred, you should have also received a return error code from Windows Installer. Look up that error code in the Windows Installer Help Library to determine the cause of the problem.</p> <p>If you did not receive a return error code from Windows Installer, this error could have been caused by the package not being authored properly. In the Windows Installer package, check to see if the AdminExecuteSequence table was defined. If that table is missing, the package cannot be decompressed.</p>

Error -9007: File With Extension Not Found

The following table documents this message:

Table 9-38 • Error -9007: File With Extension Not Found

Category	Description
Type:	Error
Message:	No file found with the extension 'ComponentKeyName'.
Cause:	This is an unexpected error that occurred when file extensions were being processed.
Resolution:	Check to make sure that the executable for the file extension exists and that it is set as the key file in its component.

Error -9008: Error Extracting Icon

The following table documents this message:

Table 9-39 • Error -9008: Error Extracting Icon

Category	Description
Type:	Error
Message:	An error occurred while extracting the icon 'IconKeyName'
Cause:	This is an unexpected error that occurred when an icon listed in the Icon table was being extracted.

Table 9-39 • Error -9008: Error Extracting Icon

Category	Description
Resolution:	Verify that the Icon entry in the Icon table is valid. If necessary, replace it with a valid icon.

Error -9009: Unknown Provider

The following table documents this message:

Table 9-40 • Error -9009: Unknown Provider

Category	Description
Type:	Error
Message:	The specified provider is unknown 'ProviderName'.
Cause:	This is an unexpected internal error.
Resolution:	Invalid data may have been modified via the Direct Editor causing this error. Delete the Release you are building, and then create a new one and rebuild.

Error -9010: Invalid Target File Name

The following table documents this message:

Table 9-41 • Error -9010: Invalid Target File Name

Category	Description
Type:	Error
Message:	The target file name is invalid. 'FileName'
Cause:	This is an unexpected internal error.
Resolution:	Invalid data may have been modified via the Direct Editor causing this error. Verify the Name field on the Citrix Assistant / ThinApp Assistant Profile Information page and make sure the name does not contain any invalid file name characters.

Error -9011: Error Reading MSI Table

The following table documents this message:

Table 9-42 • Error -9011: Error Reading MSI Table

Category	Description
Type:	Error
Message:	Unexpected error reading MSI table 'TableName'
Cause:	This is an unexpected error that occurred when the specified Windows Installer table was being processed.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9012: Unexpected Error in Method

The following table documents this message:

Table 9-43 • Error -9012: Unexpected Error in Method

Category	Description
Type:	Error
Message:	Unexpected error in method 'MethodName'
Cause:	This is an unexpected internal error.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9013: Type Library Not Found

The following table documents this message:

Table 9-44 • Error -9013: Type Library Not Found

Category	Description
Type:	Error
Message:	Type library not found: 'TypeLibraryName'
Cause:	You encounter this error when a type library file was not found when trying to extract COM information.

Table 9-44 • Error -9013: Type Library Not Found

Category	Description
Resolution:	Check to see if the type library file exists in the proper location when building the profile. If this does not resolve the problem, perform these additional investigational steps and then contact AdminStudio Technical Support.

Error -9014: ShellExecute Failed

The following table documents this message:

Table 9-45 • Error -9014: ShellExecute Failed

Category	Description
Type:	Error
Message:	ShellExecute failed: 'CommandLine'
Cause:	You encounter this error when the specified command line failed to launch a process.
Resolution:	Check to see if the executable file name shown is a valid file and that the user has the proper access rights to run it. If this does not resolve the problem, perform these additional investigational steps and then contact AdminStudio Technical Support.

Error -9015: Unable to Determine Full Path for Driver

The following table documents this message:

Table 9-46 • Error -9015: Unable to Determine Full Path for Driver

Category	Description
Type:	Warning
Message:	Unable to determine the full path for driver 'DriverName'
Cause:	You encounter this error when a driver referenced in the ODBCDataSource table is not being installed by the package.

Table 9-46 • Error -9015: Unable to Determine Full Path for Driver

Category	Description
Resolution:	<p>This error can be resolved in one of two ways:</p> <p>Editing the Windows Installer Package</p> <ol style="list-style-type: none"> 1. Edit the package using InstallShield Direct Edit Mode. 2. Navigate to the ISVirtualPackage table. 3. Create an entry as follows to identify the full path of the missing driver: <ul style="list-style-type: none"> Name: <DriverName> Description Value: Path to Driver <p>Manually Installing the Driver</p> <p>Install the missing driver on your machine and then rebuild the Citrix profile.</p>

Warning -9016: Contents of Table Ignored

The following table documents this message:

Table 9-47 • Warning -9016: Contents of Table Ignored

Category	Description
Type:	Warning
Message:	Contents of table 'TableName' will be ignored
Cause:	This error message identifies a known limitation of Citrix conversion.
Resolution:	If the contents of the table is deemed critical, repackage the application, and then rebuild the Citrix profile.

Warning -9017: .NET 1.x Assembly Not Supported

The following table documents this message:

Table 9-48 • Warning -9017: .NET 1.x Assembly Not Supported

Category	Description
Type:	Warning
Message:	Assembly 'AssemblyName' is a .NET 1.x assembly and will not be converted correctly. Only .NET 2.0/3.0 assemblies are currently supported. You may wish to repackage this package first.
Cause:	You encounter this error when attempting to convert a package containing a .NET 1.x assembly. Only .NET 2.0/3.0 assemblies are currently supported.
Resolution:	Repackage the application, and then rebuild the Citrix profile.

Warning -9018: Custom Actions Ignored

The following table documents this message:

Table 9-49 • Warning -9018: Custom Actions Ignored



Category	Description
Type:	Warning
Message:	Custom action 'CustomActionName' will be ignored.
Cause:	<p>When converting a Windows Installer package to a Citrix profile, all custom actions are ignored. Any modifications to a target machine that a custom action in this Windows Installer package may create will not be propagated into the Citrix profile.</p>  <p>Note • When a custom action that does not modify the system or perform any part of the installation (such as an InstallShield Editor predefined custom action or a Type 19 custom action) is encountered, no message is generated. If a Type 51 custom action is encountered (which sets a property from a formatted text string), it is automatically resolved. If a Type 35 custom action is encountered, it is only resolved if it is referenced in the Directory table.</p>

Table 9-49 • Warning -9018: Custom Actions Ignored

Category	Description
Resolution:	<p>The resolution that you should perform depends upon the purpose of the custom action:</p> <ul style="list-style-type: none"> • If the custom action merely automatically enters a value or makes some other kind of minor modification, you can ignore this warning. • If the custom action does something which could change the behavior of the installation (such as setting a Property), you may need to resolve this issue. <p>To resolve this issue, first attempt to launch the converted package on Citrix XenApp. If you receive any application errors, you need to repackage this application.</p>  <p>To repackage a Windows Installer package to capture custom action functionality:</p> <ol style="list-style-type: none"> 1. Use the Repackaging Wizard to repackage this application. The Repackaging Wizard monitors system changes as an application is installed, and then that data is converted into a Repackager project. 2. Build the Repackager project to generate a revised Windows Installer package. This new Windows Installer package does not contain any custom actions, but (as a result of being repackaged) it will include the functionality performed by the original custom action.

Warning -9019: Conditionalized Components

The following table documents this message:

Table 9-50 • Warning -9019: Conditionalized Components

Category	Description
Type:	Warning
Message:	There exist one or more conditionalized components which may not be converted correctly
Cause:	This warning is generated when attempting to convert conditionalized components because conditions on components are not evaluated.
Resolution:	<p>Repackage the application on a machine that has a similar environment to the machines where the profile will be deployed. Then rebuild the Citrix profile.</p> <p>You can also evaluate the conditions on the listed components and remove the components you know are not needed for your target machines. Then rebuild the Citrix profile.</p>

Error -9020: Directory With Null Parent

The following table documents this message:

Table 9-51 • Error -9020: Directory With Null Parent

Category	Description
Type:	Error
Message:	Directory 'DirectoryName' has a null parent and will be ignored.
Cause:	This error occurs if a directory table entry (other than TARGETDIR) is null.
Resolution:	Evaluate the ThinApp application to see if it works. If it does not work properly, you may want to consider repackaging the package.

Error -9021: Unable to Extract COM Data

The following table documents this message:

Table 9-52 • Error -9021: Unable to Extract COM Data

Category	Description
Type:	Error
Message:	Unable to extract COM data for 'FileName'
Cause:	<p>This Windows Installer package has an entry in the TypeLib or SelfReg table that contains COM data that AdminStudio cannot convert to application data.</p> <p>Depending upon which file cannot be COM extracted, it is possible that this application will still work properly in Citrix XenApp isolation environment if you repackage this Windows Installer package with COM table mapping turned off.</p> <p>COM data is stored in the Windows Registry. So, if you repackage this Windows Installer package, the capture process will be able to obtain all of this data because it captures all modifications to the Registry and does not have to rely on COM extraction.</p>
Resolution:	To resolve this issue, you need to repackage your Windows Installer package with COM table mapping turned off.

Error -9022: Complus Table

The following table documents this message:

Table 9-53 • Error -9022: Complus Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'Complus'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a Complus table. During the conversion process, the Complus table is not read.
Resolution:	The Complus table contains information needed to install COM+ applications. While Citrix XenApp supports communicating with COM+ applications, it does not support <i>installing</i> COM+ services. Therefore, this application cannot be deployed on Citrix XenApp.

Error -9024: FileSFPCatalog

The following table documents this message:

Table 9-54 • Error -9024: FileSFPCatalog

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'FileSFPCatalog'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a FileSFPCatalog table. During the conversion process, the FileSFPCatalog table is not read.
Resolution:	The FileSFPCatalog table associates specified files with the catalog files used by system file protection. If this file is necessary for the function of the application, you need to use Repackager to repackage the application.

Warning -9026: LaunchCondition Table

The following table documents this message:

Table 9-55 • Warning -9026: LaunchCondition Table

Category	Description
Type:	Warning
Message:	The conversion process does not support data in the MSI table 'LaunchCondition'.
Cause:	You encounter this warning when the Windows Installer package that you are converting includes a LaunchCondition table. During the conversion process, the LaunchCondition table is not read.
Resolution:	<p>The LaunchCondition table contains a list of conditions that all must be satisfied for the installation to begin. For example, if an application requires Windows XP to run, Windows XP is listed in the LaunchCondition table. Because this table is not read, no check is performed. Therefore, when a user on an operating system other than Windows XP launches this Citrix profile, the application may not function properly.</p> <p>To resolve this issue, perform one of the following tasks:</p> <ul style="list-style-type: none">• Option 1: Set Requirements on the Profile Requirements Page—If the launch conditions only include operating system, service pack, and language requirements, open this package in the InstallShield Editor Citrix Assistant / ThinApp Assistant, and set those Operating System and Language requirements on the Profile Requirements page. Then deploy this application as a Citrix profile.• Option 2: Determine if Launch Conditions are Met—Review the launch conditions listed in the table, and determine if the desktops in your enterprise meet those requirements. If all of the desktops meet those requirements, you can deploy this application as a Citrix profile. <p>If the desktops do not meet those requirements (such as having Internet Explorer 6.0 instead of 7.0), upgrade those desktops to meet these requirements, and then deploy this application as a Citrix profile.</p>

Warning -9027: LockPermissions Table

The following table documents this message:

Table 9-56 • Warning -9027: LockPermissions Table

Category	Description
Type	Warning
Message:	The conversion process does not support data in the MSI table 'LockPermissions'.

Table 9-56 • Warning -9027: LockPermissions Table

Category	Description
Cause:	You encounter this warning when the Windows Installer package that you are converting includes a LockPermissions table. During the conversion process, the LockPermissions table is not read.
Resolution:	<p>The LockPermissions table is used to secure individual portions of your application (files, registry keys, and created folders) in a locked-down environment.</p> <p>Citrix does not support permissions on files, registry keys, or created folders. You cannot modify permissions on any of the application's ACLs (access control lists). Because users will have full permissions when running this application in the isolation environment, this warning will not result in any problems.</p>

Error -9028: MoveFile Table

The following table documents this message.

Table 9-57 • Error -9028: MoveFile Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'MoveFile'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a MoveFile table. During the conversion process, the MoveFile table is not read.
Resolution:	<p>This MoveFile table contains a list of files to be moved or copied from a specified source directory to a specified destination directory. Because this table is not read, you need to do one of the following to resolve this issue:</p> <ul style="list-style-type: none"> • Option 1: Edit the Windows Installer Package—Open the Windows Installer package in InstallShield Editor and modify it to eliminate the use of the MoveFile table by installing additional files in the specified directories. • Option 2: Repackage the Application—Use the Repackaging Wizard to repackage this application, and then build the Repackager project to generate a revised Windows Installer package. • Option 3: Write a Pre-Launch Script—Write a pre-launch script that performs the file moving operations identified in the MoveFile table upon application launch.

Error -9029: MsiDriverPackages Table

The following table documents this message:

Table 9-58 • Error -9029: MsiDriverPackages Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'MsiDriverPackages'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a MsiDriverPackages table. During the conversion process, the MsiDriverPackages table is not read.
Resolution:	<p>The MsiDriverPackages table includes one record for each driver package component in the application.</p> <p>Citrix XenApp does not support any type of driver. For example, when installing a printer, you can install the printer software within the isolation environment, but not the printer drivers.</p> <p>Therefore, to resolve this issue, you need to install any required drivers outside of the isolation environment on the user desktop machines.</p>

Warning -9030: ODBCTranslator Table

The following table documents this message:

Table 9-59 • Warning -9030: ODBCTranslator Table

Category	Description
Type:	Warning
Message:	The conversion process does not support data in the MSI table 'ODBCTranslator'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ODBCTranslator table. During the conversion process, the ODBCTranslator table is not read.
Resolution:	<p>The ODBCTranslator table lists the ODBC translators belonging to the installation. ODBC translators translate one form of raw data into another form that can be used with a specific database type.</p> <p>Ignoring the ODBCTranslator table should not prevent your application from functioning properly. However, if it does, you need to use Repackager to repackage the application.</p>

Warning -9031: RemoveFile Table

The following table documents this message:

Table 9-60 • Warning -9031: RemoveFile Table

Category	Description
Type:	Warning
Message:	The conversion process does not support data in the MSI table 'RemoveFile'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a RemoveFile table. During the conversion process, the RemoveFile table is not read. This warning is displayed only if the application installation removes files during install (not uninstall).
Resolution:	<p>The RemoveFile table contains a list of files to be removed. If this file removal requirement is just a clean-up step, that does not impact the function of the application, you do not need to address this issue.</p> <p>However, if the existence of the files listed in the RemoveFile table prevents the application from functioning, you need to set the isolation option of the files to Ignore so that they are not visible to the isolation environment. The Ignore option directs the isolation environment to always look for the file on the system (ignoring the one inside the isolation environment).</p>

Warning -9032: RemoveIniFile Table

The following table documents this message:

Table 9-61 • Warning -9032: RemoveIniFile Table

Category	Description
Type:	Warning
Message:	The conversion process does not support data in the MSI table 'RemoveIniFile'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a RemoveIniFile table. During the conversion process, the RemoveIniFile table is not read.
Resolution:	The RemoveIniFile table contains the information an application needs to delete from a .ini file. If the removal of this entry is necessary for the function of the application, you need to use Repackager to repackage the application.

Warning -9033: RemoveRegistry Table

The following table documents this message:

Table 9-62 • Warning -9033: RemoveRegistry Table

Category	Description
Type:	Warning
Message:	The conversion process does not support data in the MSI table 'RemoveRegistry'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a RemoveRegistry table. During the conversion process, the RemoveRegistry table is not read.
Resolution:	<p>The RemoveRegistry table contains the registry information the application needs to delete from the system registry. If this removal requirement is just a clean-up step, that does not impact the function of the application, you do not need to address this issue.</p> <p>However, if the existence of the registry keys listed in the RemoveRegistry table prevents the application from functioning, you need to set the isolation option of the registry keys to Ignore so that they are not visible to the isolation environment. The Ignore option directs the isolation environment to always look for the registry key on the system (ignoring the one inside the isolation environment).</p>

Error -9036: ISCEInstall Table

The following table documents this message:

Table 9-63 • Error -9036: ISCEInstall Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISCEInstall'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISCEInstall table. During the conversion process, the ISCEInstall table is not read.
Resolution:	The ISCEInstall table is used to install Windows mobile applications. The conversion of mobile applications to Citrix profiles is not supported.

Error -9037: ISComPlusApplication Table

The following table documents this message:

Table 9-64 • Error -9037: ISComPlusApplication Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISComPlusApplication'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISComPlusApplication table. During the conversion process, the ISComPlusApplication table is not read.
Resolution:	The ISComPlusApplication table contains information about COM+ applications. While Citrix XenApp supports communicating with COM+ applications, it does not support <i>installing</i> COM+ services. Therefore, this application cannot be deployed on Citrix XenApp.

Error -9038: ISPalmApp Table

The following table documents this message:

Table 9-65 • Error -9038: ISPalmApp Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISPalmApp'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISPalmApp table. During the conversion process, the ISPalmApp table is not read.
Resolution:	The ISPalmApp table is used to install Palm mobile applications. The conversion of mobile applications to Citrix profiles is not supported.

Error -9039: ISSQLScriptFile Table

The following table documents this message:

Table 9-66 • Error -9039: ISSQLScriptFile Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISSQLScriptFile'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISSQLScriptFile table. During the conversion process, the ISSQLScriptFile table is not read.
Resolution:	<p>The ISSQLScriptFile table lists SQL scripts. When a Windows Installer package is installed, it can run an SQL script to update a database. An application running as a Citrix profile cannot update a database.</p> <p>To resolve this issue, you need to update the database prior to using the converted Citrix profile using one of the following methods:</p> <ul style="list-style-type: none">• Use scripts to update the database manually.• Update it by running the Windows Installer installation on one of the machines in your network that has access to that database.

Error -9040: ISVRoot Table

The following table documents this message:

Table 9-67 • Error -9040: ISVRoot Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISVRoot'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISVRoot table. During the conversion process, the ISVRoot table is not read.
Resolution:	<p>The ISVRoot table installs a Web site. An application running as a Citrix profile in an isolation environment cannot create a Web site. Therefore, creating Citrix profiles for applications that create Web sites during installation is not supported.</p>

Error -9041: ISXmlFile Table

The following table documents this message:

Table 9-68 • Error -9041: ISXmlFile Table

Category	Description
Type:	Error
Message:	The conversion process does not support data in the MSI table 'ISXmlFile'.
Cause:	You encounter this error when the Windows Installer package that you are converting includes a ISXmlFile table. During the conversion process, the ISXmlFile table is not read.
Resolution:	The ISXmlFile table modifies XML files. If the modification of XML files is required in order for this application to operate properly, you need to use Repackager to repackage this application.

Error -9051: Package Decompression Canceled

The following table documents this message:

Table 9-69 • Error -9051: Package Decompression Canceled

Category	Description
Type:	Error
Message:	Package decompression canceled by the user
Cause:	The user cancelled the process of decompression of compressed MSI packages.

Error -9100: CreateInstance of Package Object Failed

The following table documents this message:

Table 9-70 • Error -9100: CreateInstance of Package Object Failed

Category	Description
Type:	Error
Message:	CreateInstance of the Citrix package object failed.
Cause:	Unexpected internal error.

Table 9-70 • Error -9100: CreateInstance of Package Object Failed

Category	Description
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9101: Create Operation of Package Object Failed

The following table documents this message:

Table 9-71 • Error -9101: Create Operation of Package Object Failed

Category	Description
Type:	Error
Message:	Create operation on Citrix package object failed 'ObjectName'.
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9102: Failed to Write Header Information

The following table documents this message:

Table 9-72 • Error -9102: Failed to Write Header Information

Category	Description
Type:	Error
Message:	Failed to write package header information.
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9103: Citrix Finalization Failed

The following table documents this message:

Table 9-73 • Error -9103: Citrix Finalization Failed

Category	Description
Type:	Error
Message:	Citrix Finalization Failed
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9104: Citrix Save Failed

The following table documents this message:

Table 9-74 • Error -9104: Citrix Save Failed

Category	Description
Type:	Error
Message:	Citrix Save Failed
Cause:	Unexpected internal error. This error may sometimes occur when the profile is to be digitally signed.
Resolution:	Deselect the option to digitally sign the Citrix profile and then rebuild it.

Error -9105: Error Initializing Citrix Writer

The following table documents this message:

Table 9-75 • Error -9105: Error Initializing Citrix Writer

Category	Description
Type:	Error
Message:	Unexpected error initializing Citrix writer
Cause:	Unexpected internal error.

Table 9-75 • Error -9105: Error Initializing Citrix Writer

Category	Description
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9106: Error Initializing Citrix Package

The following table documents this message:

Table 9-76 • Error -9106: Error Initializing Citrix Package

Category	Description
Type:	Error
Message:	Unexpected error initializing Citrix package
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9107: Error Writing Citrix File Entries

The following table documents this message:

Table 9-77 • Error -9107: Error Writing Citrix File Entries

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix file entries.
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9108: Error Determining Source File Path

The following table documents this message:

Table 9-78 • Error -9108: Error Determining Source File Path

Category	Description
Type:	Error
Message:	Unexpected error determining source file path for 'FileName'
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9109: Error Writing Citrix Folder Entries

The following table documents this message:

Table 9-79 • Error -9109: Error Writing Citrix Folder Entries

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix folder entries
Cause:	Unexpected internal error.
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9110: Error Writing Citrix Registry Entries

The following table documents this message:

Table 9-80 • Error -9110: Error Writing Citrix Registry Entries

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix registry entries
Cause:	Unexpected internal error.

Table 9-80 • Error -9110: Error Writing Citrix Registry Entries

Category	Description
Resolution:	First check to see if the product was installed properly. Then, perform preliminary investigational steps and contact AdminStudio Technical Support.

Error -9113: Error Writing Citrix INI File Entries

The following table documents this message:

Table 9-81 • Error -9113: Error Writing Citrix INI File Entries

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix INI file entries
Cause:	Unexpected internal error.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9114: Error Writing Citrix Shortcuts

The following table documents this message:

Table 9-82 • Error -9114: Error Writing Citrix Shortcuts

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix shortcuts
Cause:	A catastrophic error has occurred while writing shortcuts to the profile.
Resolution:	Verify that shortcuts point to a valid file. Try to narrow down issue by only keeping one shortcut and then try to rebuild.

Error -9115: Error Saving Citrix Profile

The following table documents this message:

Table 9-83 • Error -9115: Error Saving Citrix Profile

Category	Description
Type:	Error
Message:	Unexpected error saving Citrix profile
Cause:	A catastrophic error has occurred while saving the Citrix profile.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9116: Error Creating Empty Citrix Profile

The following table documents this message:

Table 9-84 • Error -9116: Error Creating Empty Citrix Profile

Category	Description
Type:	Error
Message:	Unexpected error creating empty Citrix profile
Cause:	AdminStudio is unable to create a new internal instance of a Citrix profile.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9117: Error Creating Intermediate Folder

The following table documents this message:

Table 9-85 • Error -9117: Error Creating Intermediate Folder

Category	Description
Type:	Error
Message:	Unexpected error creating intermediate folder
Cause:	AdminStudio is unable to create the intermediate folder used for the build. This error could occur if the user does not have permission to write to C:\TMP.

Table 9-85 • Error -9117: Error Creating Intermediate Folder

Category	Description
Resolution:	Obtain write access to C:\TMP and then rebuild the profile.

Error -9118: Error Initializing Citrix Profile

The following table documents this message:

Table 9-86 • Error -9118: Error Initializing Citrix Profile

Category	Description
Type:	Error
Message:	Unexpected error initializing Citrix profile.
Cause:	The initial values on the new profile could not be set.
Resolution:	Check the package name, description, version, and security settings for any possible causes.

Error -9119: Error Creating Default Target in Citrix Profile

The following table documents this message:

Table 9-87 • Error -9119: Error Creating Default Target in Citrix Profile

Category	Description
Type:	Error
Message:	Unexpected error creating default target in Citrix profile
Cause:	Initial target in the new profile could not be created.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9120: Error Deleting File From Profile

The following table documents this message:

Table 9-88 • Error -9120: Error Deleting File From Profile

Category	Description
Type:	Error

Table 9-88 • Error -9120: Error Deleting File From Profile

Category	Description
Message:	Unexpected error deleting file 'FileName' from profile
Cause:	Specified file could not be deleted from profile.
Resolution:	<p>Check to see if the file exists and if the user has access rights to the file.</p> <p>If that did not resolve the problem, perform these additional investigational steps and then contact AdminStudio Technical Support.</p>

Error -9121: Failed to Copy File into Citrix Profile

The following table documents this message:

Table 9-89 • Error -9121: Failed to Copy File into Citrix Profile

Category	Description
Type:	Error
Message:	Failed to copy file into Citrix profile. Error: 'Name' File: 'Name'
Cause:	Specified file could not be copied into profile.
Resolution:	<p>Check to see if the file exists and if the user has access rights to the file.</p> <p>Also, when this error occurred, you should have also received a return error code from Windows Installer. Look up that error code in the Windows Installer Help Library to determine the cause of the problem.</p>

Error -9122: Target Does Not Exist in Citrix Profile

The following table documents this message:

Table 9-90 • Error -9122: Target Does Not Exist in Citrix Profile

Category	Description
Type:	Warning
Message:	The target for shortcut 'ShortcutName' does not exist in the Citrix profile. Excluding shortcut.
Cause:	Actual file that shortcut points to is not included in the package.

Table 9-90 • Error -9122: Target Does Not Exist in Citrix Profile

Category	Description
Resolution:	Exclude the shortcut by clearing the selection on the Citrix Assistant Profile Shortcuts page, and then rebuild the profile.

Error -9124: No Shortcuts Created for this Profile

The following table documents this message:

Table 9-91 • Error -9124: No Shortcuts Created for this Profile

Category	Description
Type:	Error
Message:	No shortcuts were created for this profile
Cause:	A Citrix profile must include at least one valid shortcut.
Resolution:	Add a shortcut on the Citrix Assistant Profile Shortcuts page, and then rebuild the profile.

Error -9125: Error Writing Citrix File Type Associations

The following table documents this message:

Table 9-92 • Error -9125: Error Writing Citrix File Type Associations

Category	Description
Type:	Error
Message:	Unexpected error writing Citrix file type associations
Cause:	Unable to write file type associations.
Resolution:	Perform preliminary investigational steps and then contact AdminStudio Technical Support.

Error -9126: Failed to Sign Profile Using Certificate

The following table documents this message:

Table 9-93 • Error -9126: Failed to Sign Profile Using Certificate

Category	Description
Type:	Error
Message:	Failed to sign the profile using the supplied certificate
Cause:	The certificate that is being used is invalid.
Resolution:	Obtain a valid certificate and rebuild the profile.

Error -9127: Could Not Create Script Execution

The following table documents this message:

Table 9-94 • Error -9127: Could Not Create Script Execution

Category	Description
Type:	Error
Message:	Could not create script execution for 'ScriptName'
Cause:	The specified script contains invalid data.
Resolution:	<p>On the Citrix Assistant Build Settings page, delete the script from the profile, re-add it, and then rebuild the profile.</p> <p>If you are still having problems, perform these additional investigational steps and then contact AdminStudio Technical Support.</p>

Warning -9128: Duplicate Shortcut

The following table documents this message:

Table 9-95 • Warning -9128: Duplicate Shortcut

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists in the profile. Excluding duplicate shortcut.

Table 9-95 • Warning -9128: Duplicate Shortcut

Category	Description
Cause:	There are multiple shortcuts defined in this profile that refer to different Start Menu locations or to other locations in the package.
Resolution:	These shortcuts are not needed. On the Citrix Assistant Profile Shortcuts page, unselect these shortcuts, and then rebuild the profile.

Warning -9129: Duplicate Shortcut Names

The following table documents this message:

Table 9-96 • Warning -9129: Duplicate Shortcut Names

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists in the profile, but with different command line parameters. A new unique shortcut 'NewShortcutName(1)' will be created in the profile.
Cause:	There are two shortcuts defined in this profile that have the same name, even though they have different command line parameters.
Resolution:	On the Citrix Assistant Profile Shortcuts page, rename one of these shortcuts and then rebuild the profile.

Warning -9130: Duplicate Shortcut Targets

The following table documents this message:

Table 9-97 • Warning -9130: Duplicate Shortcut Targets

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists in the profile, but with different target. A new unique shortcut 'NewShortcutName(1)' will be created in the profile.
Cause:	There are two shortcuts defined in this profile that have the same name, even though they have different targets.
Resolution:	On the Citrix Assistant Profile Shortcuts page, rename one of these shortcuts and then rebuild the profile.

Warning -9131: Unable to Resolve Installer Variable

The following table documents this message:

Table 9-98 • Warning -9131: Unable to Resolve Installer Variable

Category	Description
Type:	Warning
Message:	Unable to resolve an installer variable in the string 'StringName'
Cause:	Not all Windows Installer variables can be resolved at build time. This can result in errors if your application requires a specific value.
Resolution:	Repackage this application and rebuild the profile, or use a constant value in the Windows Installer package.

Warning -9132: 16 Color Shortcut Icon Not Found

The following table documents this message:

Table 9-99 • Warning -9132: 16 Color Shortcut Icon Not Found

Category	Description
Type:	Warning
Message:	No 16 color icon found for 'ShortcutName' shortcut. Shortcut icon may look distorted when published.
Cause:	The icon used for this shortcut does not contain a 16-color image. Since Citrix currently does not support images with a higher number of colors, this icon may look distorted when shown and published in Citrix XenApp.
Resolution:	You can modify the shortcut to use a different icon or add a 16-color image to the one currently used.

Warning -9133: Shortcut Icon Not Found

The following table documents this message:

Table 9-100 • Warning -9133: Shortcut Icon Not Found

Category	Description
Type:	Warning

Table 9-100 • Warning -9133: Shortcut Icon Not Found

Category	Description
Message:	No icon found for 'ShortcutName' shortcut. Using generic Windows application icon instead.
Cause:	If no icon can be loaded for this shortcut, the generic Windows application icon is used. This can happen if the file used is corrupt or if extracting an image from it is not possible.
Resolution:	Modify the shortcut to use a different icon.

Warning -9134: Failure to Extract Icon from Executable

The following table documents this message:

Table 9-101 • Warning -9134: Failure to Extract Icon from Executable

Category	Description
Type:	Warning
Message:	Failed to extract icon from executable 'filename'. Make sure the executable is not corrupt.
Cause:	A corrupt icon file may cause this warning.
Resolution:	Modify the shortcut to use a different icon.

Error -9135: Shortcut Target is 16-Bit

The following table documents this message:

Table 9-102 • Error -9135: Shortcut Target is 16-Bit

Category	Description
Type:	Error
Message:	The target for shortcut 'ShortcutName' is 16-bit. This application may not function properly in the Citrix Isolation Environment.
Cause:	The file this shortcut points to is a 16-bit application.
Resolution:	Replace file with a newer 32-bit version. Can also test and see if the application works properly in the Citrix environment.

Warning -9136: Some Files May Not Be Decompressed

The following table documents this message:

Table 9-103 • Warning -9136: Some Files May Not Be Decompressed

Category	Description
Type:	Warning
Message:	Some files may not be decompressed from this package because it contains features with a default install level of 0.
Cause:	When installing a compressed Windows Installer package, the build engine runs an administrative installation of it to decompress it. One limitation of an administrative installation is that it does not decompress a file if the feature it is contained in has a default install level of 0. If there are any files in any components contained within those features, AdminStudio will generate an error when it attempts to copy those files into the Citrix profile, because they will not exist in the source location.
Resolution:	To resolve this issue, edit the Windows Installer package and set the default install level of that feature to something other than 0.

Warning -9137: Destination Directory Cannot Be Found

The following table documents this message:

Table 9-104 • Warning -9137: Destination Directory Cannot Be Found

Category	Description
Type:	Warning
Message:	The destination directory for the 'FileName' file cannot be found. You should consider Repackaging this application before proceeding with the conversion process.
Cause:	This is an internal error.
Resolution:	Contact AdminStudio Technical Support.

Warning -9138: Ignoring a DuplicateFile Table Entry

The following table documents this message:

Table 9-105 • Warning -9138: Ignoring a DuplicateFile Table Entry

Category	Description
Type:	Warning
Message:	Ignoring a DuplicateFile table entry because unable to resolve the property used for the DestFolder: 'INVALIDPATH'
Cause:	You might encounter this error when the Windows Installer package that you are converting includes one or more entries in the DuplicateFile table, and a property that is used in the DestFolder column for one of those entries in the DuplicateFile table cannot be resolved. For example, if the destination for a duplicate file is set by a custom action, that destination cannot be resolved during the conversion.
Resolution:	<p>The DuplicateFile table contains a list of files that need to be duplicated during installation, either to a different directory than the original file or to the same directory but with a different name. Because a destination in this table cannot be resolved, you need to do one of the following to resolve this issue:</p> <ul style="list-style-type: none">• Option 1: Edit the Windows Installer Package—Open the Windows Installer package in InstallShield and modify it to eliminate the use of the problematic entry in the DuplicateFile table by including any additional copies of that file.• Option 2: Repackage the Application—Use the Repackaging Wizard to repackage this application, and then build the Repackager project to generate a revised Windows Installer package.• Option 3: Write a Pre-Launch Script—Write a pre-launch script that—upon application launch—performs the file copy operations for the problematic entry in the DuplicateFile table.

Error -9200: ThinApp Must Be Installed

The following table documents this message:

Table 9-106 • Error -9200: ThinApp Must Be Installed

Category	Description
Type:	Error
Message:	A licensed or demo version of ThinApp must be installed on this machine in order to successfully build ThinApp applications. (www.vmware.com)
Cause:	ThinApp is not installed.

Table 9-106 • Error -9200: ThinApp Must Be Installed

Category	Description
Resolution:	Install ThinApp.

Warning -9201: Extension for Shortcut Files Must Be “.exe”

The following table documents this message:

Table 9-107 • Warning -9201: Extension for Shortcut Files Must Be “.exe”

Category	Description
Type:	Warning
Message:	The extension for the target for shortcut 'ShortcutName' is not '.exe'. Excluding shortcut.
Cause:	Shortcuts that do not have a filename extension of .exe are excluded.
Resolution:	No action is required.

Error -9202: No Applications Were Created

The following table documents this message:

Table 9-108 • Error -9202: No Applications Were Created

Category	Description
Type:	Error
Message:	No applications were created.
Cause:	Either the Windows Installer package had no shortcuts or all shortcuts were excluded.
Resolution:	Make sure that the source Windows Installer .msi package has at least one shortcut to an .exe file.

Error -9203: ThinApp Tool is Missing

The following table documents this message:

Table 9-109 • Error -9203: ThinApp Tool is Missing

Category	Description
Type:	Error
Message:	ThinApp: 'ToolName' was not found
Cause:	One of the ThinApp tools required to build a ThinApp application was not found.
Resolution:	Reinstall ThinApp.

Error -9204: Duplicate Shortcut

The following table documents this message:

Table 9-110 • Error -9204: Duplicate Shortcut

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists. Excluding duplicate shortcut.
Cause:	The source package has two shortcuts that both point to the same .exe target.
Resolution:	No action is required.

Error -9205: Identically-Named Shortcut Already Exists, But With Different Parameters

The following table documents this message:

Table 9-111 • Error -9205: Identically-Named Shortcut Already Exists, But With Different Command Line Parameters

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists, but with different command line parameters. A new, unique shortcut will be created.

Table 9-111 • Error -9205: Identically-Named Shortcut Already Exists, But With Different Command Line Parameters

Category	Description
Cause:	Two shortcuts in the package differed in arguments only.
Resolution:	No action is required.

Error -9206: Identically-Named Shortcut Already Exists, But With a Different Target

The following table documents this message:

Table 9-112 • Error -9206: Identically-Named Shortcut Already Exists, But With a Different Target

Category	Description
Type:	Warning
Message:	'ShortcutName' shortcut already exists, but with a different target. A new, unique shortcut will be created.
Cause:	Two shortcuts differed in the target pointed to only.
Resolution:	No action is required.

Error -9207: Error During Build Process (vregtool.exe)

The following table documents this message:

Table 9-113 • Error -9207: Error During Build Process (vregtool.exe)

Category	Description
Type:	Error
Message:	An error occurred during the ThinApp build process (vregtool.exe).
Cause:	An unexpected error occurred while running the vregtool.exe step of the ThinApp build process.
Resolution:	The cause of this error may be discernible by the progress messages that were displayed just before this error occurred. Also, make sure none of the files/folders in the build folder hierarchy are locked.

Error -9208: Error Occurred During Build Process (vftool.exe)

The following table documents this message:

Table 9-114 • Error -9208: Error Occurred During Build Process (vftool.exe)

Category	Description
Type:	Error
Message:	An error occurred during the ThinApp build process (vftool.exe)
Cause:	An unexpected error occurred while running the vftool.exe step of the ThinApp build process.
Resolution:	The cause of this error may be discernible by the progress messages that were displayed just before this error occurred. Also, make sure none of the files/folders in the build folder hierarchy are locked.

Error -9209: Error Occurred During Build Process (tlink.exe)

The following table documents this message:

Table 9-115 • Error -9209: Error Occurred During Build Process (tlink.exe)

Category	Description
Type:	Error
Message:	An error occurred during the ThinApp build process (tlink.exe)
Cause:	An unexpected error occurred while running the tlink.exe step of the ThinApp build process.
Resolution:	The cause of this error may be discernible by the progress messages that were displayed just before this error occurred. Also, make sure none of the files/folders of the build folder hierarchy are locked.

Error -9300: Unhandled Exception During AdviseFile Operation

The following table documents this message:

Table 9-116 • Error -9300: Unhandled Exception During AdviseFile Operation

Category	Description
Type:	Error

Table 9-116 • Error -9300: Unhandled Exception During AdviseFile Operation

Category	Description
Message:	An unhandled exception occurred during the AdviseFile operation for rule 'RuleName'
Resolution:	Contact AdminStudio Technical Support.

Error -9301: Unhandled Exception During AdviseRegistry Operation

The following table documents this message:

Table 9-117 • Error -9301: Unhandled Exception During AdviseRegistry Operation

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the AdviseRegistry operation for rule 'RuleName'
Resolution:	Contact AdminStudio Technical Support.

Error -9302: Unhandled Exception During Command Action

The following table documents this message:

Table 9-118 • Error -9302: Unhandled Exception During Command Action

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the command action with the description 'CommandActionName'
Resolution:	Contact AdminStudio Technical Support.

Error -9303: Unhandled Exception During Alter File Action

The following table documents this message:

Table 9-119 • Error -9303: Unhandled Exception During Alter File Action

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the alter file action with the description 'FileName'
Resolution:	Contact AdminStudio Technical Support.

Error -9304: Unhandled Exception During Alter Registry Action

The following table documents this message:

Table 9-120 • Error -9304: Unhandled Exception During Alter Registry Action

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the alter registry action with the description 'RegistryName'
Resolution:	Contact AdminStudio Technical Support.

Error -9305: Unhandled Exception During Create Action

The following table documents this message:

Table 9-121 • Error -9305: Unhandled Exception During Create Action

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the create action with the description 'CreateName'
Resolution:	Contact AdminStudio Technical Support.

Error -9306: Unhandled Exception During Execution of Rules Engine

The following table documents this message:

Table 9-122 • Error -9306: Unhandled Exception During Execution of Rules Engine

Category	Description
Type:	Error
Message:	An unhandled exception occurred during the execution of the rules engine.
Resolution:	Contact AdminStudio Technical Support.

Error -9401: Error Initializing App-V Writer

The following table documents this message:

Table 9-123 • Error -9401: Error Initializing App-V Writer

Category	Description
Type:	Error
Message:	Unexpected error initializing App-V writer.
Resolution:	Contact AdminStudio Technical Support.

Error -9402: Error Initializing App-V Package

The following table documents this message:

Table 9-124 • Error -9402: Error Initializing App-V Package

Category	Description
Type:	Error
Message:	Unexpected error initializing App-V package.
Resolution:	Contact AdminStudio Technical Support.

Error -9403: Error Writing App-V File Entries

The following table documents this message:

Table 9-125 • Error -9403: Error Writing App-V File Entries

Category	Description
Type:	Error
Message:	Unexpected error writing App-V file entries.
Resolution:	Contact AdminStudio Technical Support.

Error -9404: Error Writing App-V Folder Entries

The following table documents this message:

Table 9-126 • Error -9404: Error Writing App-V Folder Entries

Category	Description
Type:	Error
Message:	Unexpected error writing App-V folder entries
Resolution:	Contact AdminStudio Technical Support.

Error -9405: Error Writing App-V Registry Entries

The following table documents this message:

Table 9-127 • Error -9405: Error Writing App-V Registry Entries

Category	Description
Type:	Error
Message:	Unexpected error writing App-V registry entries.
Resolution:	Contact AdminStudio Technical Support.

Error -9406: Error Writing App-V INI File Entries

The following table documents this message:

Table 9-128 • Error -9406: Error Writing App-V INI File Entries

Category	Description
Type:	Error
Message:	Unexpected error writing App-V INI file entries.
Resolution:	Contact AdminStudio Technical Support.

Error -9407: Error Writing App-V Shortcuts

The following table documents this message:

Table 9-129 • Error -9407: Error Writing App-V Shortcuts

Category	Description
Type:	Error
Message:	Unexpected error writing App-V shortcuts.
Resolution:	Contact AdminStudio Technical Support.

Error -9408: Error Writing App-V File Type Data

The following table documents this message:

Table 9-130 • Error -9408: Error Writing App-V File Type Data

Category	Description
Type:	Error
Message:	Unexpected error writing App-V file type data.
Resolution:	Contact AdminStudio Technical Support.

Error -9409: Error Saving App-V Data

The following table documents this message:

Table 9-131 • Error -9409: Error Saving App-V Data

Category	Description
Type:	Error
Message:	Unexpected error saving App-V data.
Resolution:	Contact AdminStudio Technical Support.

Error -9410: Error Determining Source File Path

The following table documents this message:

Table 9-132 • Error -9410: Error Determining Source File Path

Category	Description
Type:	Error
Message:	Unexpected error determining source file path for 'FileName'.
Cause:	The installation location of a file, which is determined by some run time property, cannot be determined by the App-V virtual converter.
Resolution:	Locate the file in InstallShield and provide a known directory.

Error -9411: OSD File Template Could Not Be Extracted

The following table documents this message:

Table 9-133 • Error -9411: OSD File Template Could Not Be Extracted

Category	Description
Type:	Error
Message:	The Microsoft App-V OSD file template could not be extracted. The OSD file generation will not operate properly.
Resolution:	Contact AdminStudio Technical Support.

Error -9412: OSD File Could Not Be Saved

The following table documents this message:

Table 9-134 • Error -9412: OSD File Could Not Be Saved

Category	Description
Type:	Error
Message:	The Microsoft App-V OSD file could not be saved. The OSD file generation will not operate properly.
Resolution:	Contact AdminStudio Technical Support.

Error -9413: App-V OSD Save

The following table documents this message:

Table 9-135 • Error -9413: App-V OSD Save

Category	Description
Type:	Error
Message:	The Microsoft App-V OSD file could not be saved. The OSD file generation will not operate properly.
Resolution:	Contact AdminStudio Technical Support.

Warning -9414: Local App-V Application Specified as a Dependency of the Primary Application

The following table documents this message:

Table 9-136 • Warning -9414: Local App-V Application Specified as a Dependency of the Primary Application

Category	Description
Type:	Warning
Message:	A local App-V application was specified as a dependency of the primary application. The primary application may not run correctly if it is relocated to a different location.
Cause:	The user specified a dependent App-V application that is either on the local drive or on a mapped drive. This is determined by examining the HREF attribute of the CODEBASE tag in the dependency application's OSD file.

Table 9-136 • Warning -9414: Local App-V Application Specified as a Dependency of the Primary Application

Category	Description
Resolution:	Dependency applications should be referenced by a portable mechanism using either a non-FILE protocol or by using a network URL.

Error -9415: Dependency Application Was Not Found

The following table documents this message:

Table 9-137 • Error -9415: Dependency Application Was Not Found

Category	Description
Type:	Error
Message:	Dependency application was not found: 'ApplicationName'.
Cause:	A specified App-V dependency application file was not found.
Resolution:	Check the path of the specified App-V dependency application.

Warning -9416: Invalid Primary Application Directory

The following table documents this message:

Table 9-138 • Warning -9416: Invalid Primary Application Directory

Category	Description
Type:	Error
Message:	The specified Primary Application Directory, 'DirectoryName', does not exist.
Cause:	This may be caused if the directories specified in the Windows Installer package have changed after a valid Primary Application Directory was specified.
Resolution:	Specify a valid Primary Application Directory using the supplied browse folder in InstallShield.

Error -9417: Dependency Application's OSD File Contains an Invalid HREF Value

The following table documents this message:

Table 9-139 • Error -9417: Dependency Application's OSD File Contains an Invalid HREF Value

Category	Description
Type:	Error
Message:	Dependency application OSD file contains an invalid value for the HREF field of the CODEBASE tag: 'HREF_Field_Value'
Cause:	The CODEBASE tag of the dependency application's OSD file may have an empty or non-existent HREF attribute.
Resolution:	Make sure that the CODEBASE tag of the dependency application's OSD file has a valid HREF attribute.

Error -9418: Error While Privatizing Side-By-Side Assemblies

The following table documents this message:

Table 9-140 • Error -9418: Error While Privatizing Side-By-Side Assemblies

Category	Description
Type:	Error
Message:	An error occurred while privatizing Side-By-Side assemblies.
Cause:	When converting to an App-V package, files installed to the win32 Sxs assembly cache need to be privatized so that the App-V runtime can find them. This error occurs if there was an unexpected failure in that process.
Resolution:	Contact AdminStudio Technical Support.

Error -9419: Error Inserting Watermark

The following table documents this message:

Table 9-141 • Error -9419: Error Inserting Watermark

Category	Description
Type:	Error

Table 9-141 • Error -9419: Error Inserting Watermark

Category	Description
Message:	An error has occurred inserting the evaluation watermark into the App-V Package.
Resolution:	Contact AdminStudio Technical Support.

Warning -9500: Shortcut Missing

The following table documents this message:

Table 9-142 • Warning -9500: Shortcut Missing

Category	Description
Type:	Warning
Message:	The target for shortcut 'FileName' does not exist. Excluding shortcut.
Cause:	The target file of a shortcut in the project does not exist.
Resolution:	Repackage this application and then rebuild the virtual package.

Error -10000: Process Cancelled By User

The following table documents this message:

Table 9-143 • Error -10000: Process Cancelled By User

Category	Description
Type:	Error
Message:	Process cancelled by user.
Cause:	User clicked Cancel to cancel the profile conversion process.

Warning -10001: Suite File Missing

The following table documents this message:

Table 9-144 • Warning -10001: Suite File Missing

Category	Description
Type:	Warning
Message:	The suite MSI file 'FileName' is missing and will be excluded from the conversion.
Cause:	An MSI file that is part of a suite conversion was not found.
Resolution:	Make sure the input file for the suite conversion process exists.

Warning -10002: Suite File is Duplicate

The following table documents this message:

Table 9-145 • Warning -10002: Suite File is Duplicate

Category	Description
Type:	Warning
Message:	The suite MSI file 'FileName' appears to be the same as the main MSI file and we will exclude this file from the conversion process.
Cause:	A suite conversion was attempted where the main Windows Installer file (.msi) and one of the additional Windows Installer files specified were the same.
Resolution:	Specify unique Windows Installer files as part of the suite conversion process.

Warning -10003: Application File Missing

The following table documents this message:

Table 9-146 • Warning -10003: Application File Missing

Category	Description
Type:	Error
Message:	Application file not found 'ApplicationName'

Table 9-146 • Warning -10003: Application File Missing

Category	Description
Cause:	A file referenced by the installation was not found by the App-V virtual converter. It is likely that the file reference is broken in the installation.
Resolution:	Use InstallShield to locate the file in the installation and either fix the link or delete it.

Warning -10004: INI File Missing

The following table documents this message:

Table 9-147 • Warning: -10004: INI File Missing

Category	Description
Type:	Error
Message:	INI file not found 'INI_File_Name'.
Resolution:	Contact AdminStudio Technical Support.

Fix 11000: Excluding TCPIP Registry Entries

The following table documents this message:

Table 9-148 • Fix 11000: Excluding TCPIP Registry Entries

Category	Description
Type:	Fix
Message:	Excluding TCPIP registry entries from the Citrix profile.
Action:	Windows Installer Converter will exclude all TCPIP registry entries from the Citrix profile.

Fatal Error 11001: Fail on VMware

The following table documents this message:

Table 9-149 • Fatal Error 11001: Fail on VMware

Category	Description
Type:	Fatal
Message:	VMware cannot be virtualized.
Cause:	Conversion will fail when the application being virtualized is VMware.
Action:	This error message is displayed: VMware cannot be virtualized.

Warning 11003: Control Panel Applet - Citrix

The following table documents this message:

Table 9-150 • Warning 11003: Control Panel Applet - Citrix

Category	Description
Type:	Warning
Message:	The Control Panel Applet [AppletName] will not appear in Control Panel.
Action:	Windows Installer Converter will display a warning when the application contains a control panel applet.

Fix 11004: Control Panel Applet - ThinApp

The following table documents this message:

Table 9-151 • Fix 11004: Control Panel Applet - ThinApp

Category	Description
Type:	Fix
Message:	Generating shortcut for the Control Panel Applet located at 'DirectoryPath'
Action:	Windows Installer Converter will create a default shortcut for ThinApp Control Panel applets.

Fatal Error 11005: QuickTime 7.4.1 Causes Fatal Error

The following table documents this message:

Table 9-152 • Error 11005: QuickTime 7.4.1 Causes Fatal Error

Category	Description
Type:	Fatal Error
Message:	QuickTime 7.4.1 is known to have errors when running from a virtual package. Use QuickTime 7.4.5 instead.
Cause:	QuickTime 7.4.1 cannot be virtualized correctly.
Resolution:	Obtain QuickTime 7.4.5 and repeat the conversion process.

Fix 11006: Adobe Distiller Exclude AdobePDFSettings

The following table documents this message:

Table 9-153 • Fix 11006: Adobe Distiller Exclude AdobePDFSettings

Category	Description
Type:	Fix
Message:	Excluding the registry key Software\Adobe\Acrobat Distiller\AdobePDFSettings. Adobe Distiller will recreate these settings on first use.
Action:	Windows Installer Converter will exclude the AdobePDFSettings registry settings.

Fix 11007: Exclude URL Shortcut

The following table documents this message:

Table 9-154 • Fix 11007: Adobe Distiller Exclude AdobePDFSettings

Category	Description
Type:	Fix
Message:	Excluding shortcut to .URL file. App-V does not launch these files properly.
Action:	Windows Installer Converter will exclude the shortcut to the .URL file.

Steps to Take Before Calling Technical Support

Before contacting AdminStudio Technical Support, perform the following steps to attempt to clearly identify the problem you are having:

- **Check package**—To determine if this error is caused by a problem with the specific package you are converting, try to build a Citrix profile of a simple package that contains only one file.
- **Check machine and OS**—To determine if this error is caused by a configuration on a particular machine or operating system, attempt to build this Citrix profile on another machine or operating system.
- **Check individual files**—To determine if this is error limited to a specific item, find out if removing or excluding a particular item will build error free.

Application Features Requiring Pre- or Post-Conversion Actions

Some application features are ignored when creating a Citrix profile. Therefore, some additional pre- or post-conversion actions must be taken in order for the application profile to run on Citrix XenApp.

One action you could take to try to include ignored features in a Citrix profile is to first repackage the application using the Repackaging Wizard, and then convert the repackaged application to a Citrix profile.

The following table lists the application features which require additional, manual conversion steps:

Table 9-155 • Application Features Ignored During Profile Conversion



Windows Installer Feature	Manual Conversion Steps
User-Defined Custom Actions	<p>When converting a Windows Installer package to a Citrix profile, all custom actions are ignored. For user-defined custom actions, a warning message is generated. Any modifications to a target machine that a custom action in this Windows Installer package may create will not be propagated into the Citrix profile.</p> <p>The resolution that you should perform depends upon the purpose of the custom action:</p> <ul style="list-style-type: none"> • If the custom action merely automatically enters a value or makes some other kind of minor modification, you can ignore this warning. • If the custom action does something which could change the behavior of the installation (such as setting a Property), you may need to resolve this issue. <p>To resolve this issue, first attempt to launch the converted package on Citrix XenApp. If you receive any application errors, you need to repackage this application, by performing the following steps.</p>  <p>To successfully convert a package with user-defined custom actions:</p> <ol style="list-style-type: none"> 1. Use the Repackaging Wizard to repackage this application. The Repackaging Wizard monitors system changes as an application is installed, and then that data is converted into a Repackager project. 2. Build the Repackager project to generate a revised Windows Installer package. This new Windows Installer package does not contain any custom actions, but (as a result of being repackaged) it will include the functionality performed by the original custom action.
Services	<p>Citrix XenApp does not support any type of services. Therefore, to resolve this issue, you need to install any required services outside of the isolation environment on the user desktop machines.</p>  <p>To successfully convert a package with services:</p> <ol style="list-style-type: none"> 1. If you have an application and a service bundled in the same Windows Installer package, you need to create a separate Windows Installer package containing just the service. 2. Install the service on each of the user desktop machines. The Citrix profile of this application should now be able to run in an isolation environment on machines that already have the service installed.

Table 9-155 • Application Features Ignored During Profile Conversion

Windows Installer Feature	Manual Conversion Steps
COM+	While Citrix XenApp supports communicating with COM+ applications, it does not support <i>installing</i> COM+ services. Therefore, an application that contains COM+ services cannot be deployed on Citrix XenApp.

Chapter 9: Converting Windows Installer Packages to Virtual Applications

Windows Installer Converter Reference

Creating Customized Virtual Applications



Edition • The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.

You can use AdminStudio to create customized virtual applications in the Microsoft App-V, VMware ThinApp, and Citrix XenApp virtual application formats.

Information about creating virtual applications is organized into the following sections:

- [About Virtualization](#)
- [About the AdminStudio Virtualization Interface](#)
- [Creating Microsoft App-V Applications](#)
- [Creating ThinApp Applications](#)
- [Creating Citrix Profiles](#)



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

About Virtualization



Note • This section provides a description of virtualization in general for those that are not familiar with it. It does not represent the architecture of any specific vendor.

Virtualization enables you to isolate an application in its own environment so that it does not conflict with existing applications or modify the underlying operating system.

- Limitations of a Standard Installation Environment
- Benefits of Application Virtualization

Limitations of a Standard Installation Environment

A typical Windows application has dependencies on components that are shared by multiple applications. Applications access these shared system resources, such as the registry or Windows system files. When an installation author recognizes that their application references a shared system component, they include a merge module to install that component.

When one of these shared components is installed during an application's installation, it is possible that a previously installed version of the same component could be overwritten, causing the existing application to break. Because of these possible problems, extensive compatibility testing needs to be performed before an application can be distributed in the enterprise environment.

The following diagram provides an example of two conflicting installed applications.

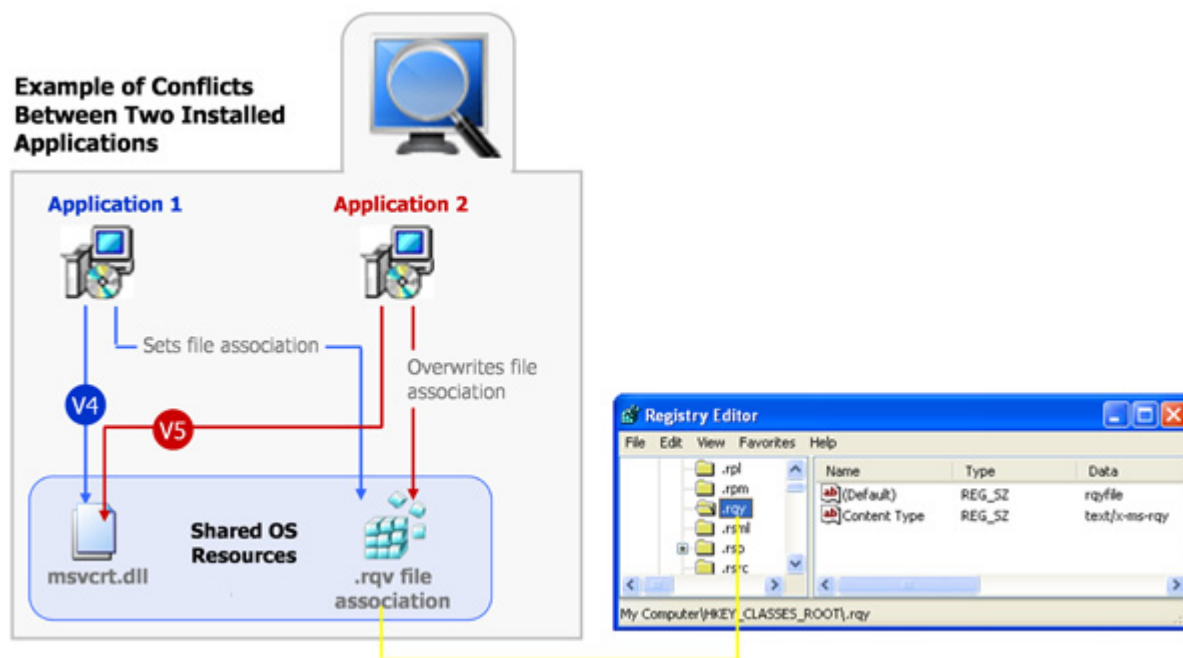


Figure 10-1: Example of Conflicts Between Two Installed Applications

Benefits of Application Virtualization

Virtual applications run in virtual environments that keep the application layer and the operating system layer separate. Each application includes its own configuration information in its virtual environment. As a result, many applications can run side-by-side with other applications on the same computer without any conflicts.

Even though virtual applications are not installed on the local machine, they exhibit the same performance, functionality, and access to local services as a locally installed application would have.

The following diagram provides an example of how application virtualization would solve the conflicts shown in the previous example.

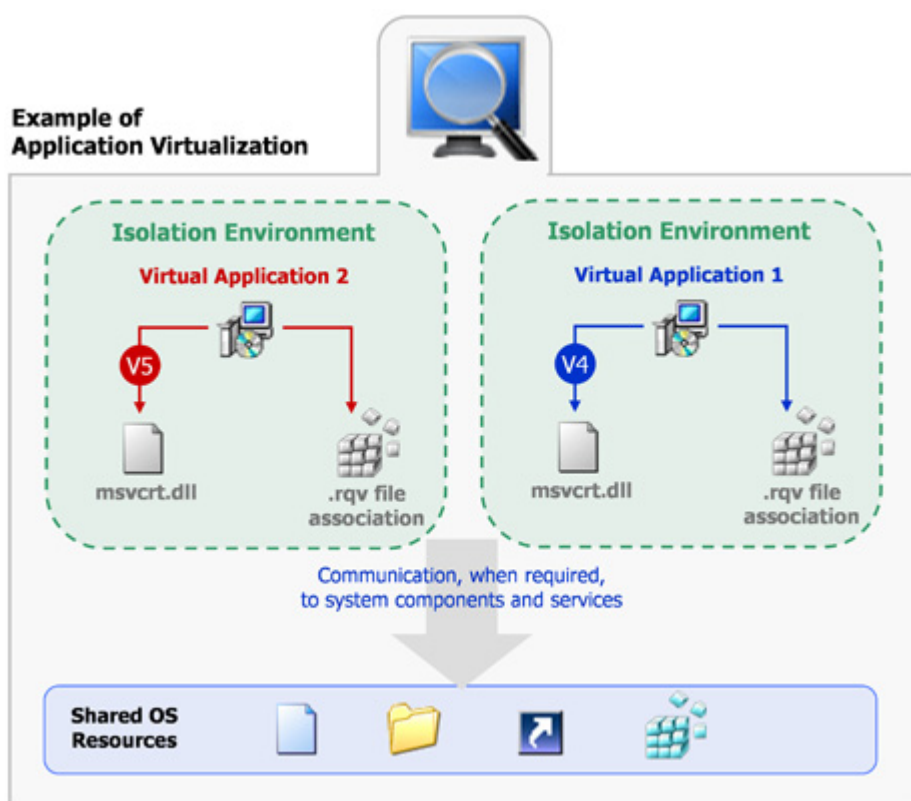


Figure 10-2: Example of Application Virtualization

Application virtualization allows the configuration of an application to be standardized to an isolation environment, rather than to an individual user's desktop machine. Application objects, files, and registry settings are contained within this isolation environment. Critical application resources are managed locally by the isolation environment, thus minimizing resource dependencies between applications.

Application virtualization greatly reduces the scope for conflicts between applications and, therefore, simplifies compatibility testing.

About the AdminStudio Virtualization Interface



Edition • *The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.*



Project • *The Microsoft App-V, ThinApp, and Citrix Assistants are available in the following project types:*

- *Basic MSI*
- *MSI Database (Direct Edit Mode)*
- *Transform (Direct MST Mode)*

AdminStudio provides the Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant to help you author a virtual application. You cannot configure a virtual application's options using the Installation Designer.

Information about the interface of these Assistants is organized in the following topics:

- [About the Virtualization Assistant Tabs](#)
- [Using the More Options, Other Places, and Help Links Sections in a Virtualization Assistant](#)
- [Navigating in a Virtualization Assistant](#)
- [Opening the Installation Designer](#)
- [Showing or Hiding the Virtualization Assistants](#)

About the Virtualization Assistant Tabs



Edition • *The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.*

When you create a new Basic MSI or MSI Database project, the **Microsoft App-V**, **VMware ThinApp**, and **Citrix XenApp** tabs are displayed in the AdminStudio interface. The home page of each Assistant has a diagram that illustrates the process of creating a virtual application using that technology.

You can work within these Assistants to create a project and configure its options and requirements. You can also use the Project Assistant or the Installation Designer to define the traditional Windows Installer version of your product installation.

How the Virtualization Assistants Work

When you create a new Basic MSI or MSI Database project, the **Microsoft App-V**, **VMware ThinApp**, and **Citrix XenApp** tabs are displayed in the AdminStudio interface.

The **Project Assistant** tab and the **Installation Designer** tab show the underlying framework for your product's Windows Installer-based installation. Some of these product elements are also displayed in the virtualization Assistants, where you can configure a virtual application's options and requirements.

Integration with the Project Assistant and the Installation Designer

Information that you enter in a virtualization Assistant is saved directly to the underlying project file. The Microsoft App-V Assistant, ThinApp Assistant, Citrix Assistant, Project Assistant, and Installation Designer run simultaneously. Any changes that you make in one are reflected instantly in the other. For example, if you remove a file in one of the virtualization Assistants, that file is no longer available in your project, and it does not appear in the Project Assistant or the Installation Designer.

Using the More Options, Other Places, and Help Links Sections in a Virtualization Assistant



Edition • The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.

The left column on each page of the virtualization Assistants contains one or more lists of links to help you in creating your installation and finding information:

- **More Options**—Provides additional configuration options relating to the specific virtualization Assistant page. These are less common options that complete the functionality of the Assistant.
- **Other Places**—The view in the Installation Designer that corresponds to the current virtualization Assistant page. Clicking the link launches the full Installation Designer and activates that view.
- **Help Links**—This list provides links to help topics pertinent to the current virtualization Assistant page.

Navigating in a Virtualization Assistant



Edition • The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.



Task:

To navigate from one page of a virtualization Assistant to another, do one of the following:

- To navigate directly to a specific page, click the appropriate icon in the navigation bar at the bottom of the page.
- To follow the assistant steps sequentially, do one of the following:
 - Click the Next or Back arrow buttons to move forward or backward.
 - Press CTRL+TAB to move to the next page and CTRL+SHIFT+TAB to move to the previous page.
- To move back to the Home page and view the overview diagram, click the Home button on the navigation bar.

Opening the Installation Designer



Edition • The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.

The **Installation Designer** tab displays the views in the AdminStudio interface. You can use this tab to configure your Windows Installer package. To open a view in the Installation Designer, click the **Installation Designer** tab.



Note • The Installation Designer and the virtualization Assistants run simultaneously. Any changes that you make in one are reflected instantly in the other.

Showing or Hiding the Virtualization Assistants



Edition • The Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the Virtualization Pack.

If you want to hide the Assistant for a virtualization technology that you do not use, you can hide it so that its tab is not displayed in the InstallShield interface. Similarly, if one of the virtualization Assistants is hidden, you can choose to display it.



Task: *To show or hide a virtualization Assistant:*

On the **View** menu, click **Microsoft App-V Assistant**, **ThinApp Assistant**, or **Citrix Assistant**.

When an Assistant's command has a check mark next to it, the tab for that Assistant is shown in the InstallShield interface. When the check mark is not displayed, that Assistant is hidden.

Creating Microsoft App-V Applications



Edition • The Microsoft App-V Assistant is included in the Virtualization Pack.

Microsoft Application Virtualization (App-V) enables applications to run as network services, removing the need for local installation of the applications. Instead, only the App-V client needs to be installed on the client machines. Even though these virtual applications are never installed, they can communicate with the local operating system, middleware, plug-ins, and other applications. Using Microsoft Application Virtualization enables you to centralize the installation and management of deployed applications, and control access to applications.

Information about Microsoft App-V and creating Microsoft App-V applications is presented in the following sections:

- [Overview of Microsoft Application Virtualization and the Microsoft App-V Assistant](#)
- [Using the Microsoft App-V Assistant to Create an App-V Application](#)
- [Microsoft App-V Assistant Reference](#)



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

Overview of Microsoft Application Virtualization and the Microsoft App-V Assistant

Microsoft Application Virtualization (App-V) enables applications to run as network services, removing the need for local installation of the applications. Instead, only the App-V client needs to be installed on the client machines. Even though these virtual applications are never installed, they can communicate with the local operating system, middleware, plug-ins, and other applications. Using Microsoft Application Virtualization enables you to centralize the installation and management of deployed applications, and control access to applications.

The **Microsoft App-V Assistant**, which you can use to configure and build App-V applications, consists of the following pages:

Table 10-1 • Pages Comprising the Microsoft App-V Assistant

Page	Description
Microsoft App-V Assistant Home Page	Displays a diagram that illustrates the process of creating an App-V application.
Package Information Page	Enter the package name, enter a comment, specify any operating system requirements, and identify the deployment server.
Files Page	View existing files and folders, add and delete files, and set isolation options for selected files and folders. Isolation options specify how the virtual environment will provide access to files and folders requested by the App-V application.
Applications Page	Create, delete, include, exclude, or rename App-V application executables, which are derived from the shortcuts in its Windows Installer package.
Registry Page	Add, delete, or modify the registry settings, and set the isolation options for selected registry keys. Isolation options specify how the virtual environment will provide access to registry keys requested by the App-V application.
Dynamic Suite Composition Page	Use to control virtual application interaction between multiple App-V packages. On this page, you can select one or more packages that need to be linked to this App-V application in order for it to execute correctly.
Build Options Page	[Basic MSI Project mode] Select the releases that you want to build. [Direct Edit or Direct MST mode] To enable the Build function for an App-V application, select the Build App-V application option.

For information on Microsoft Application Virtualization and the App-V Assistant, see the following topics:

- [About Microsoft Application Virtualization \(App-V\) and the App-V Assistant](#)
- [Components of an App-V Application](#)
- [About the Microsoft App-V Assistant](#)

About Microsoft Application Virtualization (App-V) and the App-V Assistant

This section provides an overview of Microsoft Application Virtualization and its infrastructure, and explains the benefits of using the Microsoft App-V Assistant to create App-V packages:

- [Overview](#)
- [Microsoft Application Virtualization Infrastructure](#)
- [Benefits of Using the Microsoft App-V Assistant](#)

Overview

Microsoft Application Virtualization (App-V) enables applications to run as network services, removing the need for local installation of the applications. An App-V application runs in a self-contained, virtual environment. The virtual environment contains the information necessary to run the application on the client without installing the application locally. Only the App-V client needs to be installed on the client machines. Even though these virtual applications are never installed, they can communicate with the local operating system, middleware, plug-ins, and other applications.

Because App-V applications are not installed on the client, there is minimal impact on the host operating system or other applications. As a result, application conflicts and the need for regression testing are dramatically reduced.

Using Microsoft Application Virtualization enables you to centralize the installation and management of deployed applications, and control access to applications. The App-V client presents to the end user a list of applications to which that user has access.

Microsoft Application Virtualization Infrastructure

The Microsoft Application Virtualization (App-V) infrastructure includes:

- **App-V Sequencer**—The App-V Sequencer converts application data into a format that is compatible with the App-V server and client, producing an App-V application.
- **App-V Server**—An App-V application can be placed on one or more App-V servers so that it can be streamed down to the clients on demand and cached locally.
- **App-V Client**—The App-V Client is the system component that enables the end user to interact with the App-V applications that are available on the App-V server.

Benefits of Using the Microsoft App-V Assistant

Instead of using the App-V Sequencer to create App-V applications, you can use the InstallShield Microsoft App-V Assistant, as shown in the following diagram:

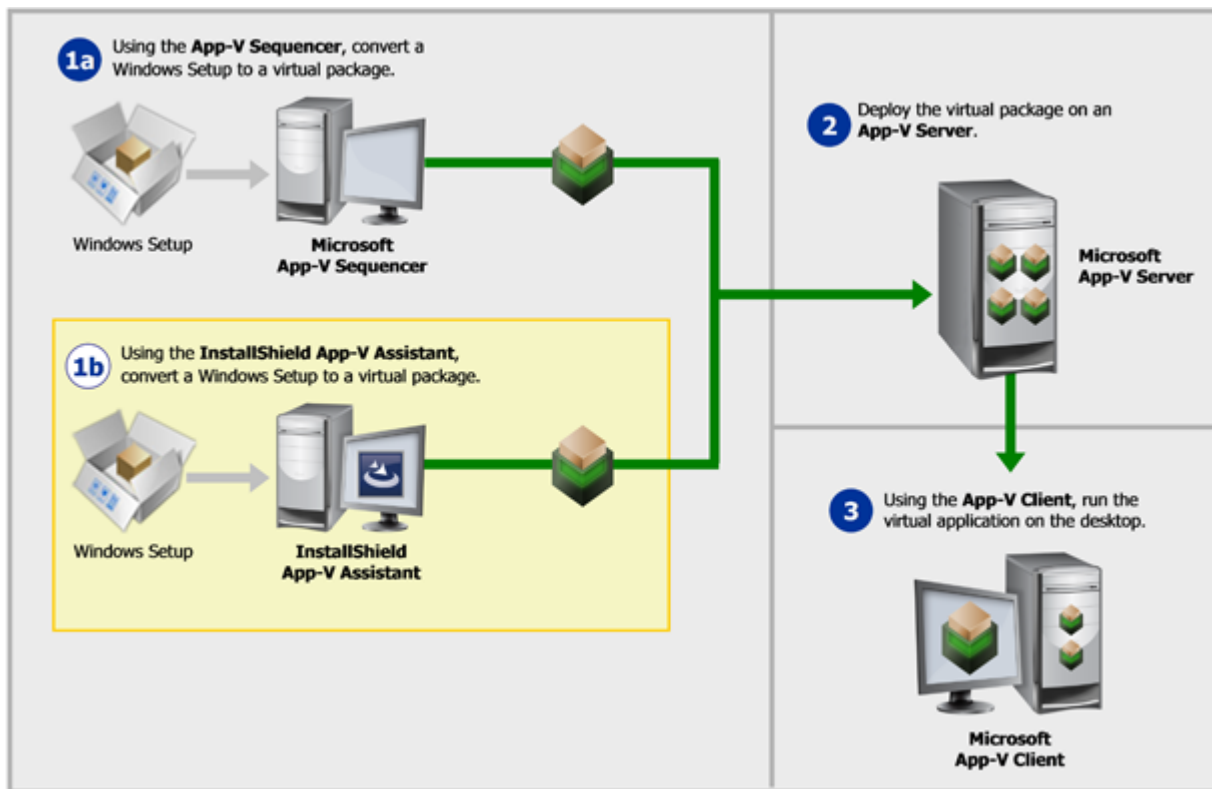



Figure 10-3: Using the App-V Assistant to Create an App-V Package

Using the App-V Assistant instead of the Microsoft App-V Sequencer to create an App-V application offers the following benefits:

Table 10-2 • Benefits of Using the App-V Assistant to Create an App-V Application

App-V Assistant Benefit	Problem	Solution
Product Installation on a Clean Machine Is Not Required	<p>The Microsoft App-V Sequencer obtains the information it needs to create an App-V application by installing a package on a clean machine and then comparing the file system snapshot that it took prior to installation with one it takes after installation. To perform this task properly, there are two requirements:</p> <ul style="list-style-type: none"> • Product must be installed on a clean machine—To ensure that all proper changes made by the installation are captured, sequencing needs to be performed on a clean machine (a computer with only the operating system, necessary service packs, and the App-V Sequencer installed on it). A new clean machine would need to be re-created for each application that is sequenced. • Installation directory must be known before sequencing can begin—In order to sequence the application effectively, you must have detailed knowledge of the how the installation is supposed to work. Prior to beginning the sequencing process, you are required to specify the installation directory for the application being sequenced. This information is often not readily available, and may require you to open the installation in an editing tool, such as InstallShield, in order to find it, or run the installation one time prior to sequencing. 	<p>Instead of installing the package, the App-V Assistant obtains the information it needs to create an App-V application directly from the installation. You are not required to have any knowledge of settings within the installation, such as the installation directory. Because there is no need to install the application to obtain this information, no permanent changes are made to the local machine and a clean machine is not required.</p>

Table 10-2 • Benefits of Using the App-V Assistant to Create an App-V Application (cont.)

App-V Assistant Benefit	Problem	Solution
Can Test the App-V Application Immediately After Conversion	To run an App-V application, the App-V Client must be installed on the machine. Because sequencing must be performed on a clean machine, which does not have the App-V Client installed, you cannot immediately test a newly created App-V application on the same machine where you sequenced it.	<p>The App-V Assistant includes a launch utility that allows you to launch and test the App-V application locally immediately after conversion before distributing it to the App-V Server.</p>  <p>Note • This feature requires that the App-V Client is installed on the local machine.</p>
Can Include Diagnostic Tools	When running a virtual application in its virtual environment, you may at some point want to examine its contents to evaluate or debug it. However, the standard diagnostic tools that you use to examine installed applications (such as the Registry Editor and the Windows Command Prompt) are not normally available within the virtual environment. When a virtual application is running within its virtual environment, applications outside of that virtual environment cannot see into it.	When you use the InstallShield App-V Assistant to create an App-V application, you can choose to include diagnostic tools with the App-V application that enable you to browse the registry and file system within the virtual environment.
Provides Easy-to-Use Interface for Dynamic Suite Composition	If you use the App-V Sequencer to create an App-V package, you have to open a separate Dynamic Suite Composition tool to edit the OSD file to create the inter-package dependencies.	The InstallShield Microsoft App-V Assistant provides a convenient, easy-to-use interface for dynamic suite composition. No separate tool is required.

Components of an App-V Application

When you use the Microsoft App-V Assistant to build an App-V virtual package, the resources you generate are called App-V applications. The components of an App-V application are shown in the following diagram:

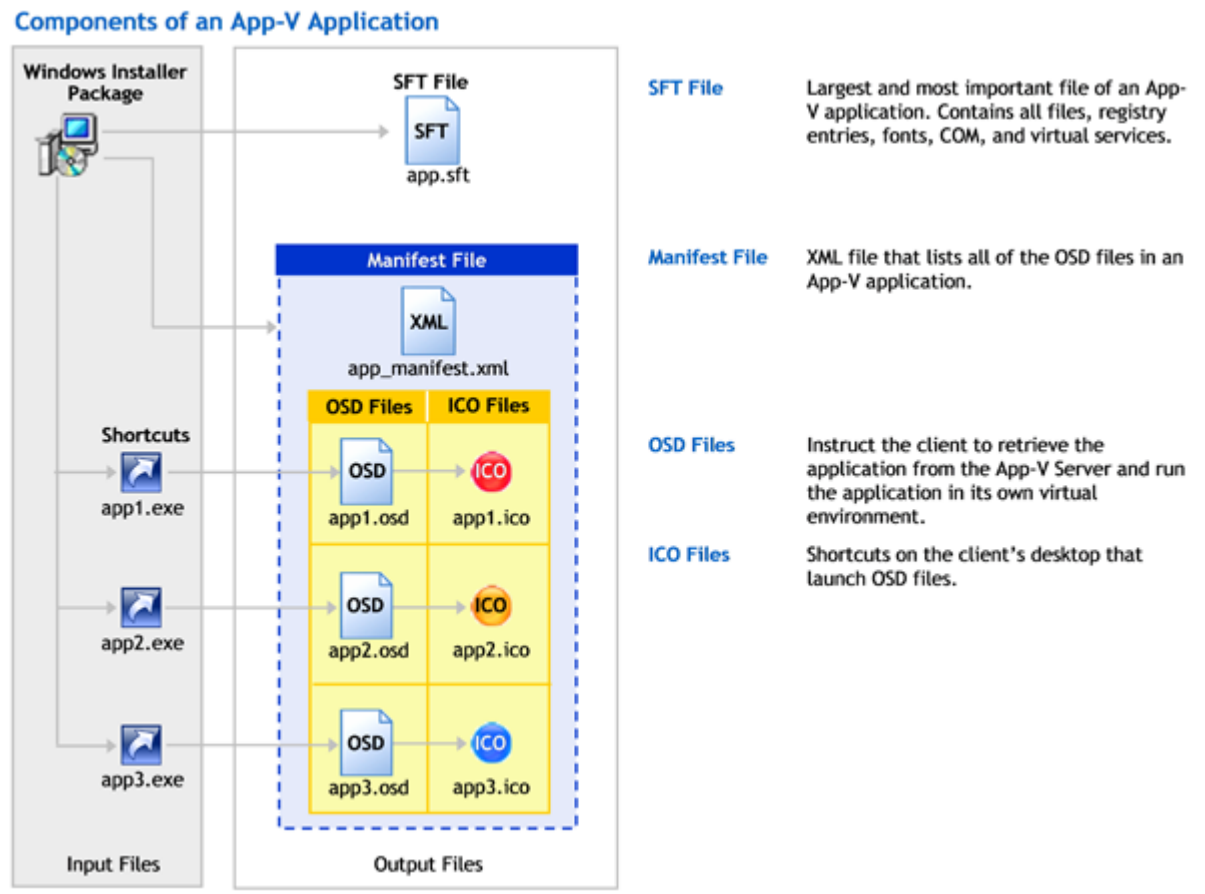


Figure 10-4: Components of an App-V Application

The following table describes the components of an App-V application:

Table 10-3 • Components of an App-V Application

File	Definition
SFT	The largest and most important file in an App-V application. All of the application's assets—including files, registry entries, fonts, COM, and virtual services—reside in this file. The maximum size of this file is 4 GB.
OSD File	An XML-based file that instructs the client about the necessary details for retrieving the App-V application from the App-V Server and running the application in its own virtual environment.
ICO File	Shortcuts on the client's desktop that point to an OSD file and launch App-V applications.
Manifest File	XML file that lists all of the OSD files in an App-V application.

Table 10-3 • Components of an App-V Application (cont.)

File	Definition
SPRJ File	App-V Sequencer project file.

When you generate an App-V application, its files are saved in the following directory path:

`App-VPackage\ProductName_vN`

The version number (*N*) of the App-V package is appended to the end of that folder path. Each time that you build an upgrade, InstallShield creates a new subfolder and increments the version number in the name of the subfolder.

The default location of the App-VPackage folder depends on whether you are building an App-V package from a Windows Installer package (.msi) or you are building an App-V package from an InstallShield project (.ism):

- If you are building an App-V package from a .msi file, InstallShield creates the App-VPackage folder in the same folder as the .msi file.
- If you are building an App-V package from an .ism file, InstallShield creates the App-VPackage folder in the following location:

`InstallShield Project Folder\project name\product configuration\release
name\DiskImages\Disk1`

A typical App-V application consists of the following files:

- `ProductName.sft`
- `ProductName.osd`
- `ProductName_manifest.xml`
- `ProductName.sprj`

InstallShield also creates a `metadata.ami` file in the same folder with the other files. This is an XML file that stores information about your product; AdminStudio uses the information in this file.

About the Microsoft App-V Assistant

Information about the Microsoft App-V Assistant is organized into the following sections:

- [Process for Authoring an App-V Application Using the Microsoft App-V Assistant](#)
- [Supported InstallShield Project Types](#)
- [How Transforms are Included in an App-V Application](#)
- [How Windows Services Are Integrated into an App-V Application](#)
- [Quick Start for Microsoft App-V Sequencer Users](#)

Process for Authoring an App-V Application Using the Microsoft App-V Assistant

You can use the Microsoft App-V Assistant to convert a Windows Installer package into an App-V application. During this process, you:

- **Package Information and Deployment Options**—Specify the package name, root folder name, enter a comment, specify any operating system requirements, and identify the deployment server.
- **Files, Folders, Shortcuts, Registry Settings**—Specify the files, folders, application shortcuts, and registry settings that will be included in the App-V application.
- **Isolation Options**—Set the isolation options for selected files, folders, and registry keys.
- **Build**—Specify build options and build an App-V application.

The following diagram illustrates the App-V application creation process:

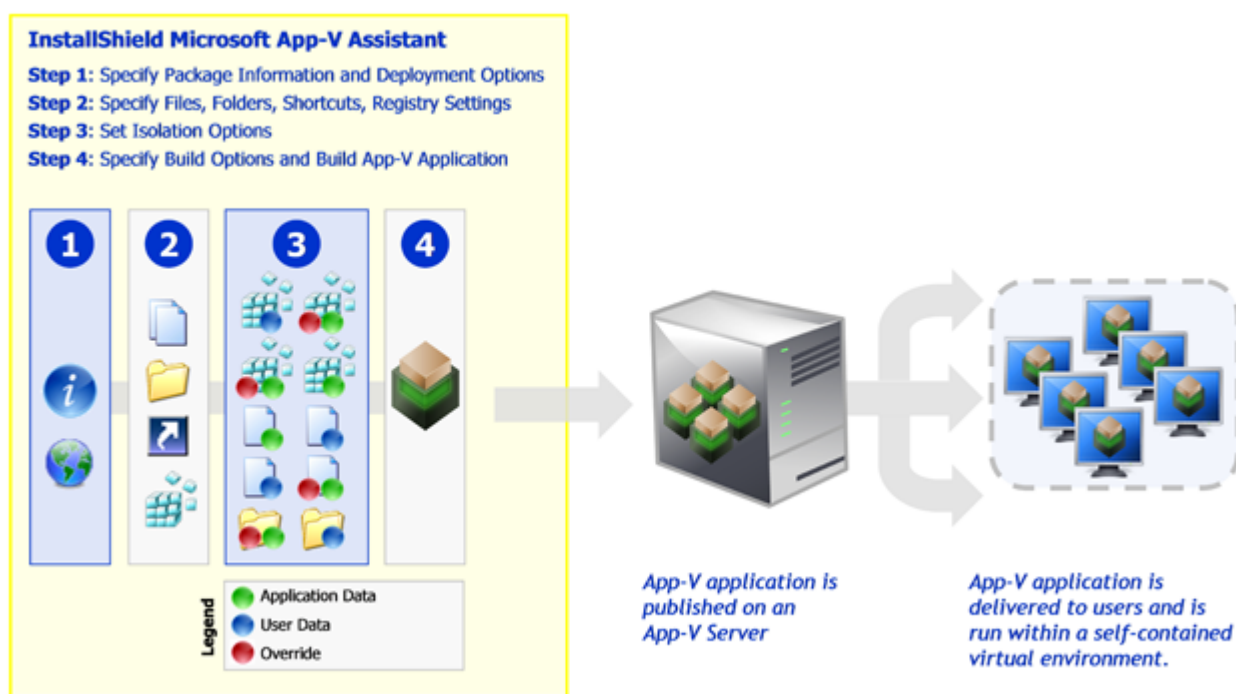


Figure 10-5: Creating an App-V Application



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

The process for authoring an App-V application using the Microsoft App-V Assistant is as follows:

Table 10-4 • Steps to Convert a Windows Installer Package to an App-V application









Step	Go To:	Actions
Getting Started	InstallShield Start Page	Create or open one of the following project types: <ul style="list-style-type: none"> • Basic MSI • MSI Database (Direct Edit Mode) • Transform (Direct MST Mode)
	InstallShield Start Page	Click on the Microsoft App-V tab to open the Microsoft App-V Assistant Home page.
	Microsoft App-V Assistant Home Page	Click Package Information in the navigation bar to open the Package Information page.
Specifying Package Information and Deployment Options	Package Information Page 	Specify the package name, root folder name, enter a comment, specify any operating system requirements, and identify the deployment server.
Managing Files in an App-V Application	Files Page 	View existing files and folders, add and delete files.
Setting Isolation Options for Folders and Files	Files Page 	Set isolation options for selected files and folders. Isolation options specify how the virtual environment will provide access to files and folders requested by the App-V application.
Modifying Shortcuts to the App-V Application's Executable Files	Applications Page 	Create, delete, include, exclude, or rename App-V application executables, which are derived from the shortcuts in its Windows Installer package.
Modifying App-V Application Registry Settings	Registry Page 	Add, delete, or modify the registry settings in your App-V application.
Setting App-V Application Registry Isolation Options	Registry Page 	Set isolation options for selected registry keys. Isolation options specify how the virtual environment will provide access to registry keys requested by the App-V application.

Table 10-4 • Steps to Convert a Windows Installer Package to an App-V application

Step	Go To:	Actions
Performing Dynamic Suite Composition	Dynamic Suite Composition Page 	Use to virtualize applications separately from the plug-ins and middleware applications that they rely on, while still enabling them to communicate with those plug-ins and middleware applications within the virtual environment.
Modifying Build Options	Build Options Page 	<p>[Basic MSI Project mode] Select the releases that you want to build.</p> <p>[Direct Edit or Direct MST mode] To enable the Build function for an App-V application, select the Build App-V application option.</p>
Building an App-V Application	Build on the Toolbar OR Build Virtual Package Button	<p>Click Build to build the active Release and create an App-V application.</p> <p>When you are in Direct Edit mode, click the Build Virtual Package button to save the Windows Installer package and create an App-V application.</p>



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

Supported InstallShield Project Types

The **Microsoft App-V** tab is available when one of the following InstallShield project types is open:

- Basic MSI Project
- MSI Database (Direct Edit Mode)
- Transform (Direct MST Mode)

How Transforms are Included in an App-V Application

The Microsoft App-V Assistant supports the inclusion of transform files with Windows Installer packages in an App-V application.

- **How transforms are applied when building an App-V application**—When building an App-V application, transforms that you have specified are automatically applied to the base Windows Installer (.msi) package to create a temporary package, and then the App-V application is generated from that temporary package.
- **Creating a new transform**—You can create a new transform in InstallShield, and then build an App-V application from that transform file. When you create a new transform file in InstallShield, you specify the root

.msi file in the **Open Transform** wizard. The steps you take to generate an App-V application after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.

- **Converting a Windows Installer package with existing transforms**—If you have a Windows Installer package and one or more existing transform files, and you want to include these transforms in the App-V application, you need to open one of the *transforms* in InstallShield (rather than the .msi file). The **Open Transform** wizard will open, and you will be prompted to specify the root .msi file and which of the existing .mst files you want to include. The steps you take to generate an App-V application after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.



Caution • All of the transforms that you add to an App-V application must be located in the same folder as the Windows Installer .msi package so that they can be accessed when the App-V application is built.

Creating Transforms

Applying Transforms

How Windows Services Are Integrated into an App-V Application

When you use the App-V Assistant to convert a Windows Installer package to an App-V application, references to Windows services that are encountered are integrated into the App-V application. In a Windows Installer package, a Windows service may be indicated by either an entry in its **ServiceInstall** table or by a Registry entry for Windows services.

- **ServiceInstall table**—If a Windows Installer package's use of a Windows service is indicated by an entry in the **ServiceInstall** table, the App-V Assistant will convert that entry to a standard Registry entry for Windows services.
- **Registry entry**—If a Windows Installer package's use of a Windows service is indicated by a Registry entry for Windows services (perhaps as the result of being repackaged), the App-V Assistant does not need to make any changes to support the application's use of the Windows service within the virtual environment.

Start Up and Shut Down Sequences

If an App-V application has an associated Windows service, App-V will start up the Windows service first, in the virtual environment, and then start up the App-V application. You will see the Windows service start up in the Task Manager as a separate process, but App-V will be running the service within the virtual environment.

Upon shut down, App-V will first shut down the App-V application and then shut down the Windows service.

Quick Start for Microsoft App-V Sequencer Users

The Microsoft App-V Assistant and the Microsoft App-V Sequencer use different directory structures for displaying files, folders, and registry entries.

- **Sequencer**—Items are shown in relation to the Primary Application Directory (also known as the mount point). See [Microsoft App-V Sequencer](#).
- **App-V Assistant**—Items are shown in relation to the Windows Installer installation directory. Also, since the location of some directories is defined during installation, installation variables are used as placeholders. See [App-V Assistant](#).

Microsoft App-V Sequencer

When viewing the files and folders of an App-V Sequencer project, they are displayed in relation to the Primary Application Directory. In the following graphic, the `myapp.exe` executable file is displayed in a subdirectory of `SoftcoMyApp.006`, the Primary Application Directory:

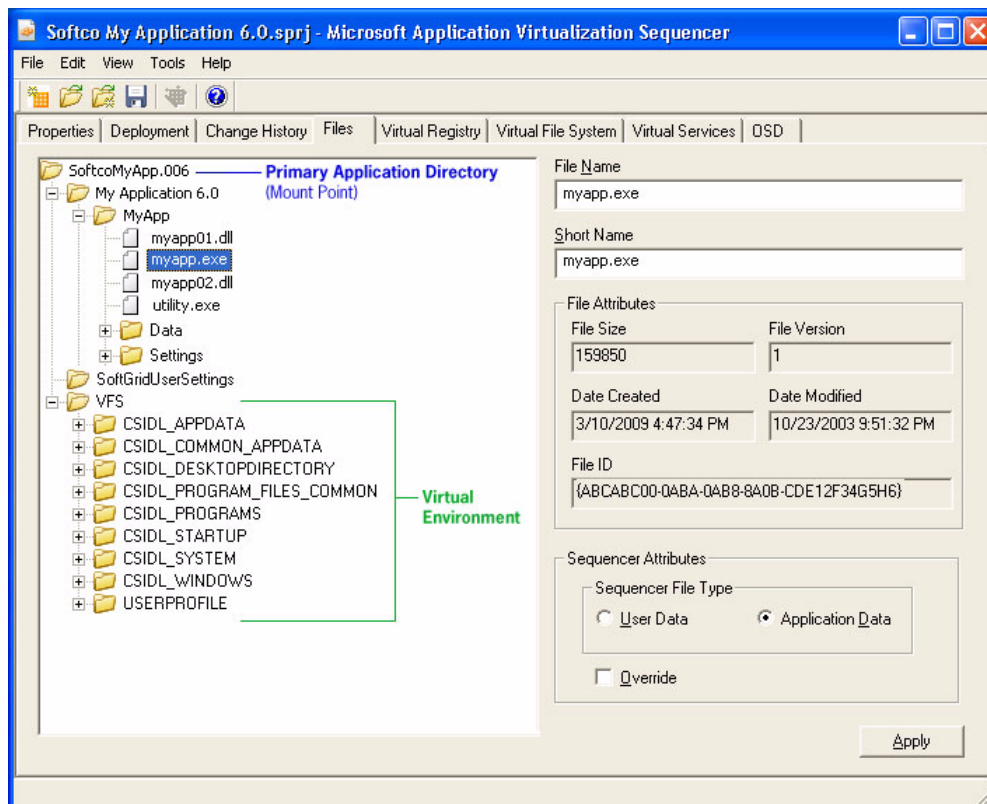


Figure 10-6: Microsoft App-V Sequencer Interface

When this App-V application is published, the `myapp.exe` executable file will be located in the following directory:

`Q:\SoftcoMyApp.006\My Application 6.0\MyApp\myapp.exe`



Note • This assumes that virtual packages are mounted on the Q: drive.

App-V Assistant

When editing a Windows Installer package or InstallShield project in the App-V Assistant, the directory structure of files, folders, and registry entries reflects the relative location of these items if they were installed. For example, on the **Files** page of the App-V Assistant, installation variables, such as `[ProgramFilesFolder]` or `[CommonFilesFolder]`, identify the installed location of their child elements.

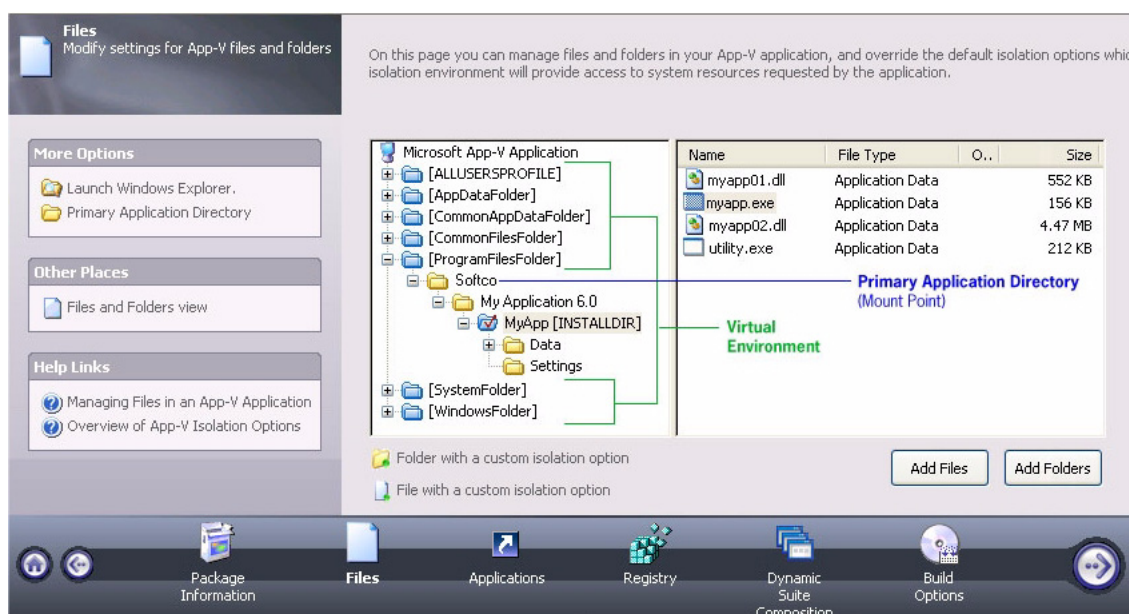


Figure 10-7: Microsoft App-V Assistant Interface

In this example, the file `myapp.exe` executable file is shown in the following directory:

`[ProgramFilesFolder]\Softco\My Application 6.0\MyApp\myapp.exe`

If this Windows Installer package were installed, the `myapp.exe` executable file would be installed in a subdirectory of the directory that is specified during installation. For example, if `C:\Program Files` is specified as the installation directory, `myapp.exe` would be installed in the following location:

`C:\Program Files\Softco\My Application 6.0\MyApp\myapp.exe`

However, when this Windows Installer package is converted to an App-V application, the build engine will map the files in the Primary Application Directory to the mount point. So,

C:\Program Files\Softco\My Application 6.0\MyApp\myapp.exe

becomes:

Q:\SoftcoMyApp.006\My Application 6.0\MyApp\myapp.exe

Also, all other references to this file—such as those in the Registry, INI files, and in Window Services—are rewired to point to the actual file location in the App-V package.



Note • This assumes that virtual packages are mounted on the Q: drive.

Comparison of Sequencer to App-V Assistant

To enable you to more clearly compare the Microsoft App-V Sequencer to the App-V Assistant, the following figure shows the location of selected folders and files in both interfaces.

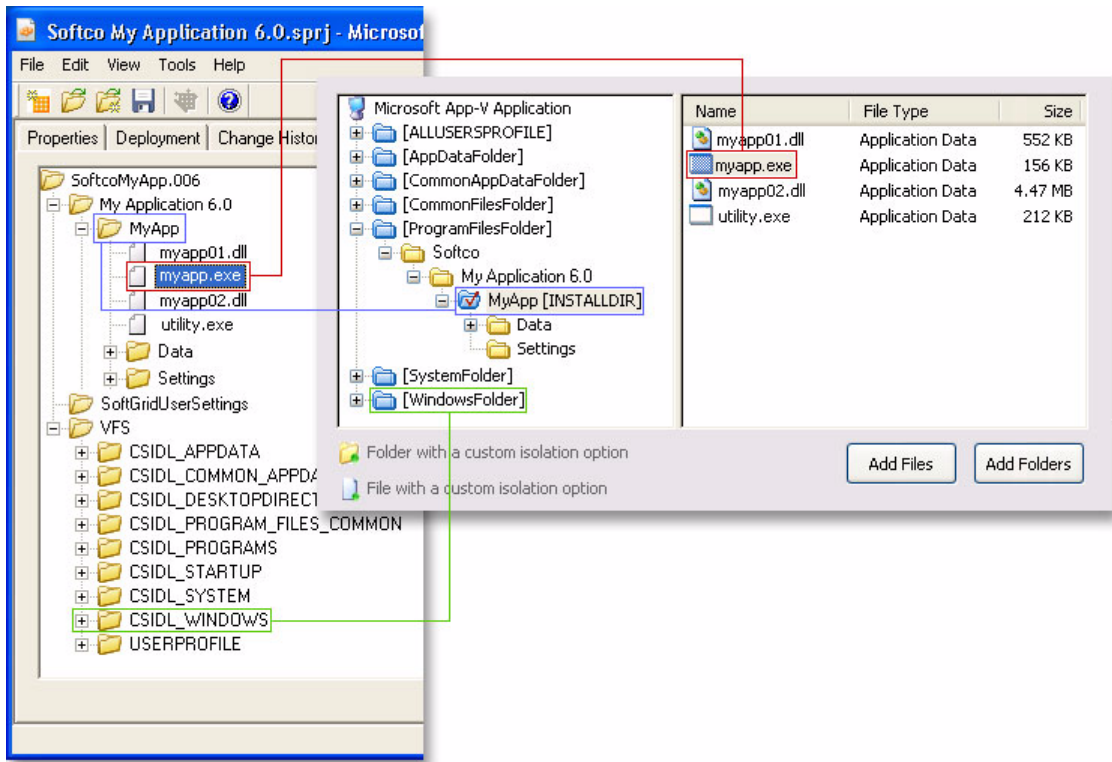


Figure 10-8: Comparison of Sequencer Interface to Microsoft App-V Assistant Interface

Using the Microsoft App-V Assistant to Create an App-V Application

The steps you need to take to create an App-V application are the following:

Table 10-5 • Steps to Take to Create an App-V Application Using the App-V Assistant

Step #	Description
Step 1	Specifying Package Information and Deployment Options
Step 2	Managing Files in an App-V Application
Step 3	Setting Isolation Options for Folders and Files
Step 4	Modifying Shortcuts to the App-V Application's Executable Files
Step 5	Modifying App-V Application Registry Settings
Step 6	Setting App-V Application Registry Isolation Options
Step 7	Performing Dynamic Suite Composition
Step 8	Modifying Build Options
Step 9	Building an App-V Application
Step 10	Testing an App-V Application Using the App-V Application Launcher

Specifying Package Information and Deployment Options

When you are creating an App-V application, the first step is to specify the package name, root folder name, and enter a comment on the Package Information page. On this page, you can also specify any operating system requirements, identify the deployment server, and specify whether to include diagnostic tools with the virtual package. This page also lets you specify upgrade information for your App-V package if appropriate.

Specifying Package Information

The first step in creating an App-V application is to enter the package name, root folder name, and a comment. To specify package information for an App-V application, perform the following steps:



Task: *To specify package information:*

1. In the **Microsoft App-V Assistant**, open the **Package Information** page.
2. In the **Package Name** field, enter a name (a maximum of 64 characters) to identify this virtual package.
3. In the **Root Folder Name** field, enter a name to identify the directory that will contain the App-V application's .sft file. When an App-V application is configured on a client machine, it is mounted on the computer's Q:\ drive in the folder you specify in this field.
 - **Default value**—By default, the **Root Folder Name** is based on the [ProductName] and [ProductVersion] properties of the App-V application's associated Windows Installer file using the 8.3 file naming convention. For example:

If ProductName is ...	And ProductVersion is ...	Then default Root Folder Name will be ...
MyApplication	1.12.3.1	MyApp1ic.112
MyApp	1	MyApp.1
MyBlueApp	1.2.3.4	MyBlueAp.123

- **Conflicts caused by duplicate values**—If you happen to have another App-V application that has identical values for the [ProductName] and [ProductVersion] properties, you can avoid conflicts by entering a unique value in the **Root Folder Name** field.
4. In the **Comments** field, enter a description of the purpose of this App-V application and any special information that you wish to document.

Specifying Operating System Requirements

To specify operating system requirements for an App-V application, perform the following steps:



Task: *To specify operating system requirements:*

1. In the **Microsoft App-V Assistant**, open the **Package Information** page.
2. Set the **Does your App-V application have any specific operating system requirements?** option to one of the following:
 - **No**—Select this option if this application will run on all of the listed operating systems. When this option is selected, the operating system check boxes are locked and cannot be changed.
 - **Yes**—Select this option if the application does not support one of the listed operating systems. When you select this option, the check boxes are unlocked and you can clear the selection of the unsupported operating systems.
3. If you selected **Yes**, select the operating systems that this application supports, and clear those that this application does not support.

Specifying Upgrade Package Information

InstallShield enables you to specify whether you want to create an upgrade for your App-V package. If you specify that you do want to create an upgrade, you can specify additional information about the upgrade, such as whether to append the version number to the App-V package file name.



Task: *To specify upgrade information:*

1. In the **Microsoft App-V Assistant**, open the **Package Information** page.
2. In the **More Options** area, click **Upgrade Settings**. The **App-V Package Upgrade Settings** dialog box opens.
3. Do one of the following:
 - To create an upgrade for your App-V package, select the **Enable Upgrade** check box. Then specify which package should be upgraded. If you want InstallShield to include the version number in the package name, select the **Append version number to package name** check box.
 - To avoid creating an upgrade package, clear the **Enable Upgrade** check box.


Specifying the Deployment Server


To specify the deployment server for an App-V application, perform the following steps:



Task: *To specify the deployment server:*

1. In the **Microsoft App-V Assistant**, open the **Package Information** page.
2. Under **Where will the App-V application be deployed?**, set the following **Server URL** options:

Option	Description
Protocol	<p>Enables you to select the protocol that will stream the sequenced application package from a virtual application server to an Application Virtualization Desktop Client. The following protocols are available:</p> <ul style="list-style-type: none">• RTSP—The default, it specifies that the Real-Time Streaming Protocol controls the exchange of virtualization-enabled applications.• RTSPS—Specifies that the Real-Time Streaming Protocol with Transport Layer Security controls the exchange of a sequenced application package.• FILE—Specifies that the sequenced application will be streamed from a file share.• HTTP—Specifies that Hypertext Transport Protocol controls the exchange of a package.• HTTPS—Specifies that Secure Hypertext Transport Protocol controls the exchange of a package.
Host	<p>Specify the virtual application server or the load balancer in front of a group of virtual application servers that will stream the software package to an Application Virtualization Desktop Client.</p> <p>You must complete this item to create a sequenced application package, but you can change from the default %SFT_SOFTGRIDSERVER% environment variable to the actual hostname or IP address of a virtual application server.</p> <div><p>Note • If you choose not to specify a static hostname or IP address, on each Application Virtualization Desktop Client you must set up an environment variable called <i>SFT_SOFTGRIDSERVER</i>.</p><ul style="list-style-type: none">• Its value must be the hostname or IP address of the virtual application server or load balancer that is this client's source of applications.• You should make this environment variable a system variable rather than a user variable.• Any Application Virtualization Desktop Client session that is running on this computer during your assignment of this variable must be closed and then opened so that the resumed session will be aware of its new application source.</div>

Option	Description
Port	<p>Specify the port on which the virtual application server or the load balancer will listen for an Application Virtualization Desktop Client's request for the package.</p> <ul style="list-style-type: none"> This information is required to create a package, but you can change it. The default port is 554.
Path	<p>Specify the relative path on the virtual application server where the software package is stored and from which it will be streamed.</p>  <p>Note • This information is required to create a package if the <code>.sft</code> file will be stored in a subdirectory of <code>CONTENT</code>; otherwise, this information is not required.</p>

Including Diagnostic Tools With an App-V Application

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **Package Information** page, you can choose to include the Registry Editor and Windows Command Prompt diagnostic tools with your App-V application.

If you include diagnostic tools with your App-V application, you will be able to look at the registry or file system for the application while it is running in its virtual environment. For example, if you were running an App-V application and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.

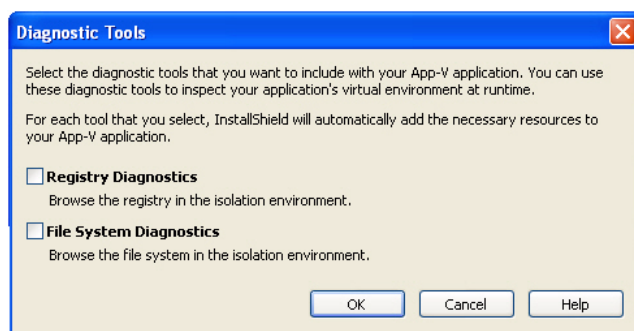


Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the App-V application. However, these tools may not be compatible with other operating systems.



Task: *To include diagnostic tools with an App-V application:*

1. In the **Microsoft App-V Assistant**, open the **Package Information** page.
2. In the **More Options** list, click **Diagnostic Tools**. The **Diagnostic Tools** dialog box opens.



3. If you want to include the Registry Editor with your App-V application so that you can browse the registry at runtime from within the virtual environment, select the **Registry Diagnostics** option.

If the **Registry Diagnostics** option is selected, a file named `Virtual Registry.osd` will be created in the App-V Package folder, which can be used to display the registry within the virtual environment. You can use `Virtual Registry.osd` to view the existing registry on the computer plus the registry for the virtual package.

4. If you want to include the Windows Command Prompt application with your App-V application so that you can browse the virtual file system at runtime from within the virtual environment, select the **File System Diagnostics** option.

If the **File System Diagnostics** option is selected, a file named `Virtual File System.osd` will be created in the App-V Package folder, which can be used to display the files and folders within the virtual environment. You can use `Virtual File System.osd` to view the existing files and folders on the computer plus the files and folders for the virtual package.

5. Click **OK**.

Launching the Diagnostic Tools Within the Virtual Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the App-V application.

When the user runs this App-V application, two additional shortcuts will be available in the application's shortcut folder: The names of these shortcuts will reflect the application name, such as:

```
[ProductName] Registry  
[ProductName] File System
```

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application's virtual environment.

Managing Files in an App-V Application

The next step in creating an App-V application is to view existing files and folders, add and delete files and folders, and set isolation options for files and folders.

The following tasks are performed on the **Files** page.

- [Adding, Deleting, and Moving Files and Folders in an App-V Application](#)
- [Controlling the Display of Predefined Folders](#)
- [Specifying the Primary Application Directory](#)

Adding, Deleting, and Moving Files and Folders in an App-V Application

The directories in the destination tree on the **Files** page of the Microsoft App-V Assistant represent how your application will look when it is installed on to your customer's machine.

On the **Files** page, you can view all of the files and folders that are currently in your App-V application, add new files and folders to include in the App-V application, and delete files and folders from the App-V application.

- [Adding Files to an App-V Application](#)
- [Adding an Existing Folder \(and its Contents\) to an App-V Application](#)
- [Creating a New Folder](#)
- [Moving Files and Folders](#)
- [Deleting Files and Folders](#)

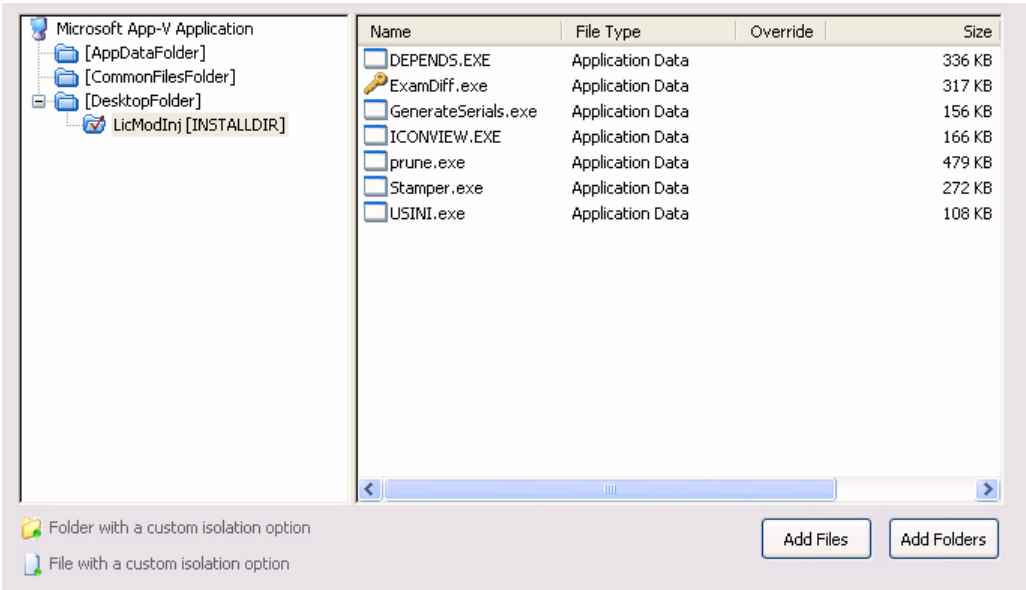
Adding Files to an App-V Application

To add files to an App-V application, perform the following steps:



Task: *To add a files to an App-V application:*

1. In the **Microsoft App-V Assistant**, open the **Files** page. The files and folders are listed in the **Microsoft App-V application** tree, organized by installation directory.



Folders are listed in the column on the left, and all of the files in the selected folder are listed on the right. Blue folders are the supported MSI standard folders. The folder with the check mark is INSTALLDIR, which represents the main product installation directory.

2. Browse through the folder tree to find the folder that you would like to add files to.
3. Select the folder and click the **Add Files** button. The **Open** dialog box opens.
4. Select the file or files that you want to add and click **Open**. The files you selected are now listed.



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Adding a File by Dragging and Dropping Files From Your System

You can also add files or folders to your App-V application on the **Files** page by dragging them from a directory on your computer to the desired location in the tree.

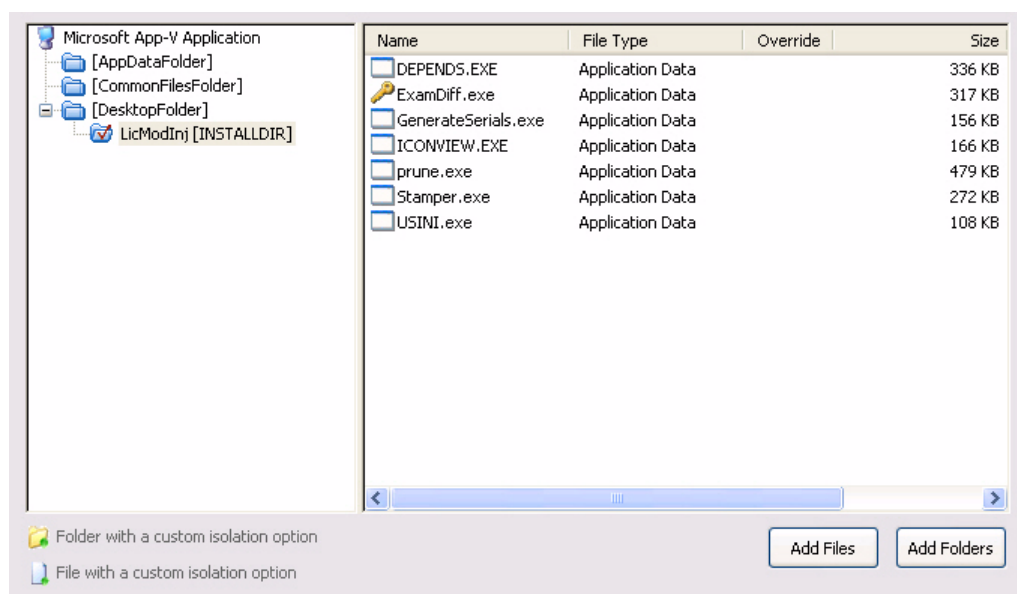
Adding an Existing Folder (and its Contents) to an App-V Application

To add an existing folder and all of the files and subfolders within it to an App-V application, perform the following steps:

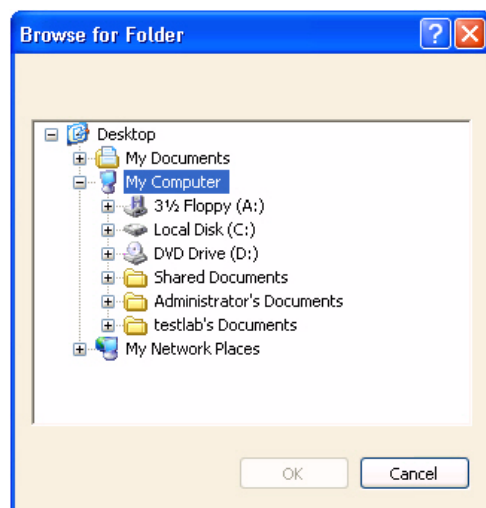


Task: *To add an existing folder to an App-V application:*

1. In the **Microsoft App-V Assistant**, open the **Files** page. The files and folders are listed in the **App-V application tree**, organized by installation directory.

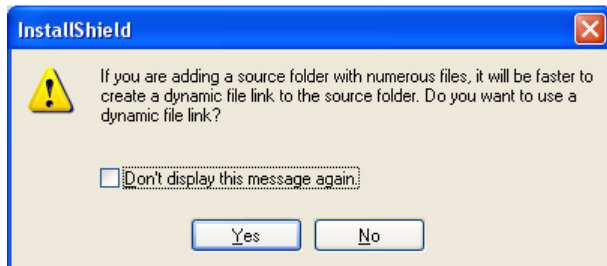


2. Browse through the folder tree to find the folder that you would like to add a folder into.
3. Select the folder and click the **Add Folders** button. The **Browse for Folder** dialog box opens, listing all of the directories available to your computer.



4. Select a folder and click **OK**.

If you are editing an InstallShield project (not a Windows Installer package), you are prompted to choose whether you want to create a dynamic file link to the source folder.



5. Indicate whether you want to create a dynamic file link by selecting one of the following:
 - **No**—For more flexibility with App-V options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
 - **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See Dynamic File Link Settings Dialog Box.

The folder that you selected is now listed, along with of the files and folders within it.

Creating a New Folder

You can create a new, empty folder by selecting an existing folder in the tree and selecting **New Folder** from the context menu.



Task: To create a new folder:

1. Right-click on a folder in the **Microsoft App-V application** tree and select **New Folder**. A new folder is created as a subfolder of the selected folder:



2. Enter a name for the new folder.

Moving Files and Folders

To change the folder's location in the App-V application folder tree structure, perform the following steps:



Task: *To move a file or folder:*

1. Select the file or folder that you want to move.
2. With the mouse button down, drag the file or folder to the new location.
3. Release the mouse button.

Deleting Files and Folders

To delete a file or a folder (and all of its contents) from an App-V application, perform the following steps:



Task: *To delete a file or folder:*

1. Select the file or folder in the **Microsoft App-V Application** tree that you want to delete.
2. Select **Delete** from the context menu. You are prompted to confirm the deletion.
3. Click **Yes**. The selected file or folder is deleted.



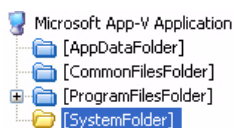
Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains from the entire Project, not just from the App-V application.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).

Controlling the Display of Predefined Folders

On the **Files** page, the **Microsoft App-V application** tree initially displays the more commonly used predefined folders, such as [CommonFilesFolder] and [ProgramFilesFolder].



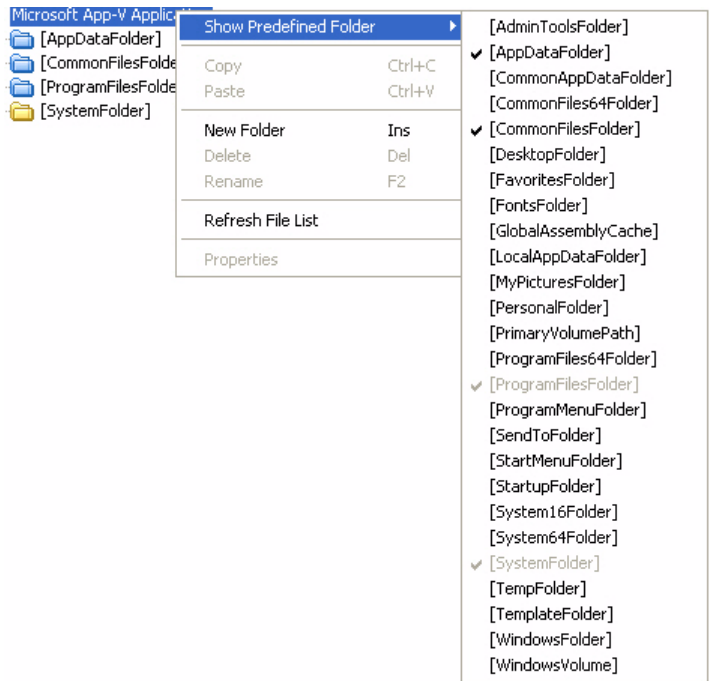
These predefined folders are dynamic, meaning that they do not use hard-coded paths. The value for each destination folder is obtained from the operating system of the target machine.

You can control which predefined folders are listed in this tree.



Task: *To change which predefined folders are listed:*

1. In the **Microsoft App-V application** tree, select the **App-V application** node (or any of the files or folders that are listed, point to **Show Predefined Folder**. A list of predefined folders opens.



Those folders that are already displayed are preceded by a check mark, and those that are not displayed do not have a check mark.

2. To add a folder to the tree listing, select a folder that is not currently listed in the tree.



Note • These predefined folders are always added to the root of the **Microsoft App-V application** tree, no matter what file or folder you had selected when you selected it from the Predefined Folders list.

3. To remove a folder from the tree listing, select that folder name in this list (which is preceded by a check mark).



Note • You cannot turn off the display of the [ProgramFilesFolder].



Specifying the Primary Application Directory

When App-V applications are run on a machine that has Microsoft Virtualization Client installed, they are run from the Client's application cache drive.

For optimum performance, it is desirable to have the bulk of the application's files mounted to this drive at run-time. In order to achieve this, it is useful to determine an application's primary application directory so that folder can be mounted to the App-V Client's application literal cache drive when the App-V application is loaded.

When an App-V application is built using InstallShield or any AdminStudio tool, the following series of steps are used to determine an App-V application's primary application directory:

Table 10-6 • Steps to Automatically Determine Primary Application Directory

#	Step	Description
1	Explicitly set primary application directory	If a directory is specified on the InstallShield App-V Assistant's Primary Application Directory dialog box (as described in Explicitly Setting the Primary Application Directory), that directory will be used.
2	Value of INSTALLDIR variable	<p>If the Windows Installer package has a value for INSTALLDIR (a system variable that specifies the root destination directory for an installation), that value will be used as the primary application directory.</p>  <p>Note • This step is not used when converting Windows Installer packages that have been repackaged using AdminStudio Repackager.</p>  <p>Note • All Windows Installer packages created by InstallShield or AdminStudio will have a value for the INSTALLDIR variable.</p>
3	Location of shortcut in a subdirectory of the ProgramFilesFolder	<p>If one of the .exe targets for a shortcut is in a subdirectory of ProgramFilesFolder, then the folder directly under ProgramFilesFolder will be used as the primary application directory. Typically, this would be:</p> <p>C:\Program Files\YourApplication</p>
4	Location of shortcut in a directory other than ProgramFilesFolder	If no .exe targets are located in a subdirectory of ProgramFilesFolder, then the target directory of a shortcut that contains an .exe will be used.
5	ProgramFilesFolder	<p>If none of the above can be found, then the primary application directory will be set to ProgramFilesFolder. Typically, this would be:</p> <p>C:\Program Files</p>

You can specify the Primary Application Directory for an App-V application by clicking **Primary Application Directory** in the **More Options** list on the **Files** page.

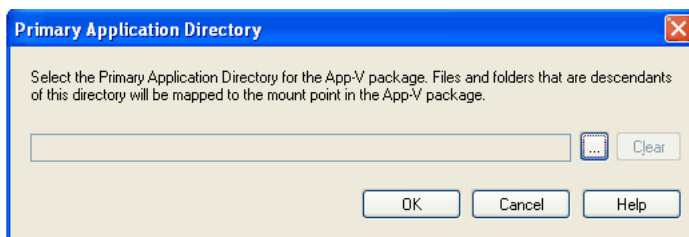
Explicitly Setting the Primary Application Directory

To specify the primary application directory for an App-V application, perform the following steps:

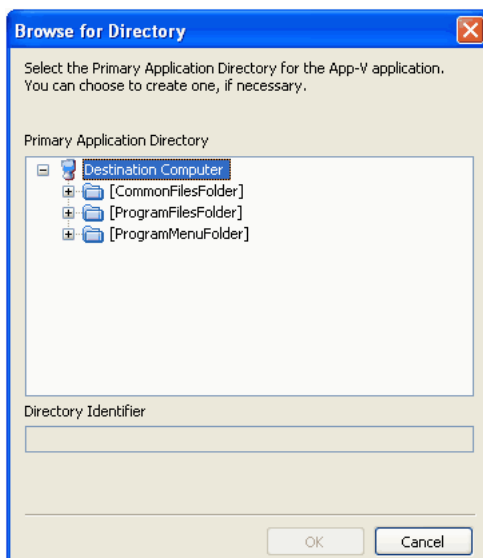


Task: *To specify the primary application directory:*

1. In the **Microsoft App-V Assistant**, open the **Files** page.
2. In the **More Options** list, click **Primary Application Directory**. The **Primary Application Directory** dialog box opens, displaying the current primary application directory setting (if one has already been specified).



3. Click the browse (..) button. The **Browse for Directory** dialog box opens, listing all of the currently available destination directories for this App-V application.



4. Select one of the listed directories and click **OK**.

Setting Isolation Options for Folders and Files

When using the Microsoft App-V Assistant, you can set isolation options to control application compatibility and accessibility. The isolation option that is assigned to a file or folder specifies how the isolation environment will provide access to system resources requested by the application.

You set isolation options on the **Isolation Options** dialog box, which is opened by selecting a file or folder and then selecting **Isolation Options** from the context menu.

Information about setting isolation options is presented in the following topics:

- [Overview of App-V Isolation Options](#)
- [Setting Isolation Options for Folders and Files](#)
- [Inheritance of Isolation Options from Folders to Files](#)



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.*

Overview of App-V Isolation Options

When using the Microsoft App-V Assistant, you can set options that uses isolation environments use to control application compatibility and accessibility. The isolation option that is assigned to a file or folder specifies how the isolation environment will provide access to system resources requested by the application.

You set isolation options on the **Isolation Options** dialog box, which is opened by selecting **Isolation Options** on the context menu when you have a file or folder selected on the **Files** page or a registry key selected on the **Registry** page.

Isolation Options for Files and Folders

The following isolation options are available:

Table 10-7 • App-V Isolation Options for Files and Folders

Option	Description
File Type	<p>When an App-V application performs a WRITE operation to a resource (a file or folder), the resource's File Type setting determines whether changes to that resource are saved for all users of the App-V application on that machine or only for the logged-in user. Select one of the following options:</p> <ul style="list-style-type: none">• Application Data—Changes to the resource are saved for all users of this App-V application on this machine.• User Data—Changes to the resource are saved only for the logged-in user.

Table 10-7 • App-V Isolation Options for Files and Folders

Option	Description
Override	<p>To have the virtualization client overwrite the resource (file or folder) when the application is upgraded and streamed from the App-V server to the client, select this check box.</p> <p>If Override check box is not selected, then the App-V client determines whether to overwrite the file or folder during an upgrade. In general, if the resource is marked as Application Data, the resource is overwritten during an upgrade; otherwise, it is not overwritten.</p>

Isolation Options for Registry Keys

The following isolation options are available:

Table 10-8 • App-V Isolation Options for Registry Keys

Option	Description
Merge with local key	Select this option to specify that the App-V application will see a merged view of the registry entries for the selected key from both the local registry and from the App-V application's registry.
Override local key	Select this option to specify that the App-V application will see only the registry entries for the selected key that are part of that App-V application.

Setting Isolation Options for Folders and Files

To set a file or folder's isolation options, perform the following steps:



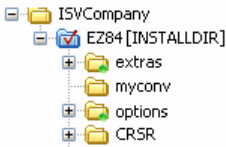
Task: *To set an isolation option on a folder or file:*

1. Open the **Files** page.
2. Browse through the folder tree to find the file or folder that you would like to modify.
3. Select the file or folder and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.

4. Specify the following options:

Option	Description
File Type	<p>When an App-V application performs a WRITE operation to a resource (a file or folder), the resource's File Type setting determines whether changes to that resource are saved for all users of the App-V application on that machine or only for the logged-in user. Select one of the following options:</p> <ul style="list-style-type: none">• Application Data—Changes to the resource are saved for all users of this App-V application on this machine.• User Data—Changes to the resource are saved only for the logged-in user.
Override	<p>To have the virtualization client overwrite the resource (file or folder) when the application is upgraded and streamed from the App-V server to the client, select this check box.</p> <p>If Override check box is not selected, then the App-V client determines whether to overwrite the file or folder during an upgrade. In general, if the resource is marked as Application Data, the resource is overwritten during an upgrade; otherwise, it is not overwritten.</p>

5. Click **OK**. Files and folders that have an isolation setting other than default are marked with a special icon:



Inheritance of Isolation Options from Folders to Files

Isolation options for files and folders are always inherited. The App-V isolation environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for C:\Windows and one for C:\Windows\System32. When the application requests C:\Windows\System32\notepad.exe, then the C:\Windows\System32 isolation rule will be applied because C:\Windows\System32 is a more specific reference to C:\Windows\System32\notepad.exe than is C:\Windows.

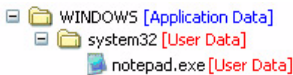


Figure 10-9: Example of Inheritance of Isolation Options from Folders to Files

Modifying Shortcuts to the App-V Application's Executable Files

You define application shortcuts to enable users to launch an App-V application from within the virtual environment.

By default, the **Microsoft App-V Assistant** creates App-V applications for all of the executable shortcuts that exist in your project (or Windows Installer package). These shortcuts are listed in a checklist on the **Applications** page.

When you select each shortcut, details about it are displayed:

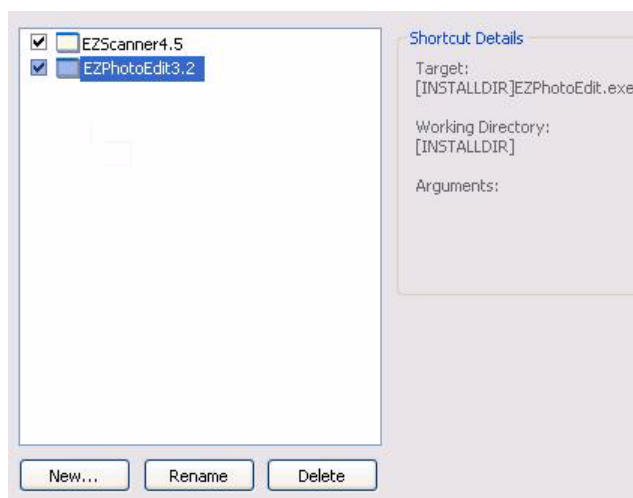


Figure 10-10: List of Shortcuts for an Application



Caution • You must define at least one shortcut to enable users to launch the application from the isolation environment.

On the **Applications** page, you can create, delete, include, exclude, or rename App-V application executables, which are derived from the shortcuts in its Windows Installer package.

- [App-V Applications and the Virtual Environment](#)
- [App-V Shortcut Requirements](#)
- [Creating a New App-V Application](#)
- [Including an Existing App-V Application](#)
- [Excluding or Deleting an Existing App-V Application](#)
- [Excluding vs. Deleting App-V Application Shortcuts](#)
- [Renaming an App-V Application](#)



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

App-V Applications and the Virtual Environment

On the **Applications** page of the Microsoft App-V Assistant, you define application shortcuts to enable users to launch an App-V application from within the virtual environment. By default, the Microsoft App-V Assistant creates App-V applications for all of the executable shortcuts that exist in your project.

To deploy an App-V application—on a local drive or a network share—systems administrators just need to give users access to the App-V application.

App-V Shortcut Requirements

For each App-V application, you are required to define **at least one** shortcut. You define application shortcuts to enable users to launch an App-V application from within the virtual environment. By default, the Microsoft App-V Assistant creates App-V applications for all of the executable shortcuts that exist in your project (or Windows Installer package).

If you build an App-V application that does not contain any shortcuts, users will not be able to launch the application.

Creating a New App-V Application

On the **Applications** page of the Microsoft App-V Assistant, you specify the executables that you want to create App-V applications for.



Task: *To create a new App-V application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project (or Windows Installer package) are listed:
 - Those that are currently included in the App-V application are selected.
 - Those that are currently excluded from the App-V application are not selected.
2. Click **New**. The **Browse for a Shortcut Target File** dialog box opens and prompts you to select a file within this App-V application.
3. Select the file that you want to create a shortcut to.
4. Click **Open**. A new shortcut is listed, and it is named the same name as the selected file.
5. To include this shortcut in the App-V application, make sure that its check box is selected.

Including an Existing App-V Application

If you want to include a previously excluded shortcut in an App-V application, perform the following steps:



Task: To include an existing App-V application:

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To include a previously excluded shortcut, select the shortcut and select the check box.

Excluding or Deleting an Existing App-V Application

By default, the Microsoft App-V Assistant creates App-V applications for all of the executable shortcuts that exist in your project (or Windows Installer package). These shortcuts are listed in a checklist on the **Applications** page.

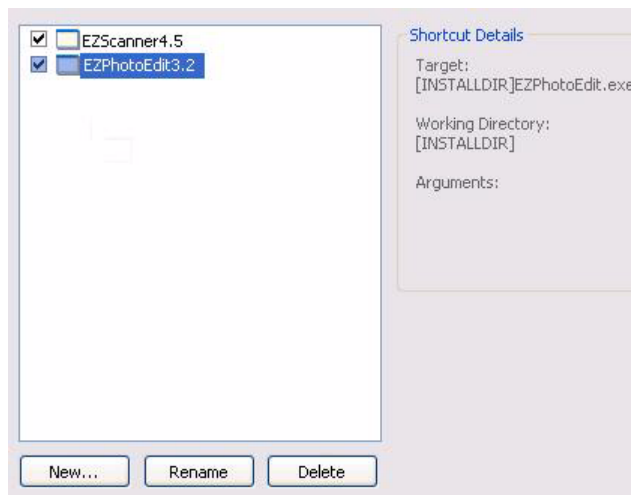


Figure 10-11: Initial List of Shortcuts for an Application

To prevent the shortcut from being created in the App-V application, you can choose to either delete or exclude it.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the App-V application, but it will remain in the InstallShield project. See [Excluding an App-V Application](#).
- **Deleting a shortcut**—When you delete a shortcut, it is removed from both the App-V application and the InstallShield project. See [Deleting an App-V Application](#).



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

If you have any unnecessary shortcuts in your project, you can simply exclude them from the App-V application by unchecking them in the shortcuts list. If you like to permanently remove a shortcut, you can delete it from the shortcut list.

Excluding an App-V Application

If you want to exclude one of these shortcuts from being created in the App-V application, perform the following steps:



Task: *To exclude an App-V application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To exclude a shortcut, select the shortcut and clear the check box.



Note • When you exclude a shortcut, it will not be created in the App-V application, but it will remain in the InstallShield project.

Deleting an App-V Application

To delete an App-V application, perform the following steps.



Task: *To delete an App-V application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
2. Select the shortcut and click **Delete**.



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

Excluding vs. Deleting App-V Application Shortcuts

To prevent a shortcut from being created in the App-V application, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the App-V application, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding an App-V Application](#).
- **Deleting a shortcut**—When you delete a shortcut, it is removed from both the App-V application and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting an App-V Application](#).

Renaming an App-V Application

To rename an App-V application, perform the following steps:



Task: *To rename an App-V application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
2. Select the shortcut that you want to rename and click **Rename**. A box appears around the shortcut name, and the shortcut name becomes an editable field.
3. Enter a new name for the shortcut.

Modifying App-V Application Registry Settings

Using the **Microsoft App-V Assistant**, you can view existing registry keys, values, and data, and add or delete registry items in your App-V application.

You can also set isolation options for selected registry keys. Isolation options specify how the isolation environment will provide access to system resources requested by the application.

Information about modifying registry settings on the **Registry** page includes the following topics:

- [About the Windows Registry](#)
- [Adding or Deleting Registry Keys and Values](#)

About the Windows Registry

The Windows registry is a system-wide database that contains configuration information used by applications and the operating system. The registry stores all kinds of information, including the following:

- Application information such as company name, product name, and version number
- Path information that enables your application to run

- Uninstallation information that enables end users to uninstall the application easily without interfering with other applications on the system
- System-wide file associations for documents created by an application
- License information
- Default settings for application options such as window positions

Keys, Value Names, and Values

The registry consists of a set of keys arranged hierarchically under the My Computer explorer. Just under My Computer are several root keys. An installation can add keys and values to any root key of the registry. The root keys that are typically affected by installations are:

- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT

A key is a named location in the registry. A key can contain a subkey, a value name and value pair, and a default (unnamed) value. A value name and value pair is a two-part data structure under a key. The value name identifies a value for storage under a key, and the value is the actual data associated with a value name. When a value name is unspecified for a value, that value is the default value for that key. Each key can have only one default (unnamed) value.

Note that the terms key and subkey are relative. In the registry, a key that is below another key can be referred to as a subkey or as a key, depending on how you want to refer to it relative to another key in the registry hierarchy.

Adding or Deleting Registry Keys and Values

Editing the registry on the **Registry** page is performed much like it is performed on the InstallShield **Registry View**. See Editing the Registry.

Setting App-V Application Registry Isolation Options

To set a registry key's isolation option, perform the following steps:



Task: *To set an isolation option on a registry key:*

1. Open the **Registry** page.
2. Browse through the registry tree to find the key that you would like to modify.
3. Select the key and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.

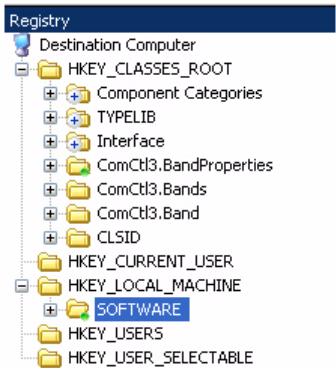


Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

4. Select one of the following options:

Option	Description
Merge with local key	Select this option to specify that the App-V application will see a merged view of the registry entries for the selected key from both the local registry and from the App-V application's registry.
Override local key	Select this option to specify that the App-V application will see only the registry entries for the selected key that are part of that App-V application.

5. Click **OK**. Registry keys that have an isolation setting of **Override local key** are marked with a special icon:



Tip • To import an existing registry (. reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.

Inheritance of Isolation Options in the Registry

Isolation options for registry keys are always inherited. The isolation environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for the `Microsoft` registry key and one for `Microsoft\Windows` registry key. When the application requests `Microsoft\Windows\CurrentVersion`, then the `Microsoft\Windows` isolation rule will be applied because `Microsoft\Windows` is a more specific reference to `Microsoft\Windows\CurrentVersion` than is `Microsoft`.

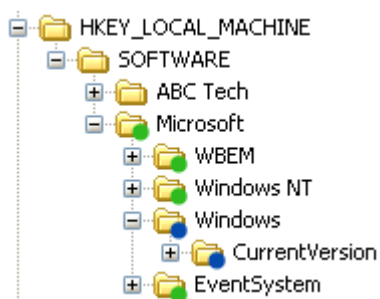


Figure 10-12: Example of Inheritance of Isolation Options from Folders to Files

Performing Dynamic Suite Composition


The point of application virtualization is to minimize the system dependencies that an application has on the underlying physical system. Many applications have common system dependencies on plug-ins or middleware, such as Adobe Reader or ODBC drivers.

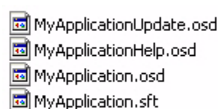
Dynamic Suite Composition (DSC) is a Microsoft Application Virtualization feature that enables applications to be virtualized separately from the plug-ins and middleware applications that they rely on, while still enabling them to communicate with those plug-ins and middleware applications within the virtual environment. The primary App-V application and the dependency App-V applications in the dynamic suite will run and interact with one another as if they were all installed locally on a computer. You would only need to deploy common system components once on each client, making them available for use by many App-V applications, rather than to include them with each of the App-V applications that are dependent upon them. This reduces redundancy in the local App-V cache and simplifies the construction and testing of the primary App-V applications.

You specify the App-V applications that you want to include in a dynamic suite on the **Dynamic Suite Composition** page of the App-V Assistant.



Task: *To compose a dynamic suite of App-V applications:*

1. In the App-V Assistant, open the **Dynamic Suite Composition** page.
2. To add a dependency App-V application, click the New () button. The **Open** dialog box opens.
3. Open the directory where the dependency App-V application that you want to add is located. That application's .osd and .sft files are listed.



4. Select one of the following:

- **One of the .osd files**—If this dependency App-V application is or is going to be published on a Server, select any one of the .osd files that are listed. If these .osd files were created properly, each of them should contain the information that will identify to the primary App-V application the published location of the dependency App-V application.



Note • It is not necessary to select more than one .osd file. All of them contain the same reference to the location of the dependency App-V application's .sft file, which is the only reference that is necessary in order for the primary App-V application to locate it.

- **The .sft file**—If this dependency App-V application is or is going to be published locally on the client or at an accessible network location, you may select just the .sft file.


The selected App-V application is now listed in the **Dependency Applications** list and, by default, the **Mandatory** option is selected.

5. Set the status of the selected App-V application by doing one of the following:

- **If the dependency App-V application is mandatory**—If the primary App-V application will not run unless it can locate this dependency App-V application, leave the **Mandatory** option selected. If a dependency App-V application that is configured as mandatory is not available, an error will be generated when someone attempts to run the primary App-V application.
- **If the dependency App-V application is not mandatory**—If the primary App-V application will run even if it cannot locate this dependency App-V application, clear the **Mandatory** option.

6. Build the primary App-V application.

Deleting a Dependency Application from the List

To remove an App-V application from the Dependency Applications list, select the application and click the Delete () button.

Modifying Build Options

On the Build Options page, you choose which releases of this InstallShield project you want to build an App-V application for when the project is built, and specify whether you want to include additional Windows Installer packages in the virtual package.

Also, if you are editing a Windows Installer package in Direct Edit mode (or Direct MST mode), you need to select the **Build App-V application** option on the Build Options page before you will be able to build an App-V application for that Windows Installer package.



Important • You must create at least one Release (on the Releases view of the Installation Designer) before you will be able to select a Release on the Build Options page.

Selecting Releases to Build

You select the releases that you want to build an App-V application for on the **Releases** tree of the **Build Options** page.



Important • You cannot create or edit a release in the Microsoft App-V Assistant. If no releases exist, you can simply click the **Build** toolbar button to create a new release or open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build an App-V application. For more information, see [Creating and Building Releases](#).

If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.



Task: To select releases to build:

1. Open the **Build Options** page.
2. Select the releases in the **Releases** tree that you want to build an App-V application for.



Important • When you select a release on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build an App-V application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and an App-V application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See [Performing Batch Builds](#).

Enabling App-V Application Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **Microsoft App-V Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build an App-V application for this Windows Installer package. To do this, perform the following steps:



Task: *To enable App-V application building when in Direct Edit Mode:*

1. Open a Windows Installer package or a transform file in InstallShield. It will be opened in Direct Edit Mode or Direct MST Mode, and the Build function (**Build** on the **Build** menu and the **Build** toolbar button) will be disabled.
2. Open the **Build Options** page of the Microsoft App-V Assistant.
3. Select the **Build App-V application** option. After you select this option, the **Build App-V application** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.

Specifying Whether to Compress the Data Files in an App-V Package

InstallShield lets you specify whether you want to use zlib compression for the data files in the App-V package.



Task: *To specify whether to compress the data files in the App-V package:*

1. Open the **Build Options** page.
2. Specify the appropriate option for the **Would you like to compress the data in the virtual package?** option.



Including Additional Windows Installer Packages in an App-V Application

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual package, you can also use the App-V Assistant to convert an application suite of multiple Windows Installer packages into one virtual package.

To include additional Windows Installer packages in an App-V application, set the **Would you like to include additional MSI files in the virtual package?** option on the **Build Options** page to **Yes**, and then select the packages that you want to add.



Task: *To include additional Windows installer packages in an App-V application:*

1. Open the **Build Options** page.
2. Set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**.
3. Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.
 - The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.
 - Use the Delete button () to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Building a Windows Installer Package to Assist in the Distribution of an App-V Application

You can choose to build a Windows Installer package to assist in the distribution of an App-V application. This simplifies the deployment of an App-V application by enabling you to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management.

To build a Windows Installer package with your App-V application, select the **Generate an installation package as part of the build output** check box on the Advanced Settings dialog box. You can access this dialog box by clicking the Advanced Settings link in the More Options area on the Build Options page of the Microsoft App-V Assistant. By default, this check box is not selected.

When you run this Windows Installer file, the minimally required App-V application files will be “installed” in the local App-V client system cache. (The .sft file remains on the App-V server until the client requests that it be downloaded when the application is launched for the first time.)



Note • The App-V client must be installed on the local machine before you can install an App-V application. The installation will detect and warn if the App-V client is not available, and the installation will fail.

To remove an installed App-V application, you need to use the Application Virtualization Client tool, which is available in the Administrative Tools of the Windows Control Panel.

Specifying Package Feature Block Optimizations

You can use the App-V Assistant **Package Optimizations** feature to control the performance and network traffic associated with running an App-V application. The package optimization option you select determines how quickly the App-V application will launch, and how often additional functionality will need to be streamed to the client while the App-V application is being used.

The files in an App-V application can be grouped into two feature blocks:

- **Feature block 1**—Feature block 1 must contain the core functionality of the App-V application that is necessary to launch the application. At application launch, all of the files in feature block 1 are streamed to the client in one unit.
- **Feature block 2**—Feature block 2 can contain additional functionality of the App-V application that is not necessary to launch the application. While the App-V application is being used, the files in feature block 2 can be streamed in small packets on an as-needed basis.

You can either choose to include all App-V application files in feature block 1 or, to improve launch speed, you can choose to group the files into two feature blocks: feature block 1 and feature block 2.

You indicate your package optimization preference on the **Package Optimizations** dialog box, which is opened by clicking the **Package Optimizations** link in the **More Options** menu of the **Build Options** page.

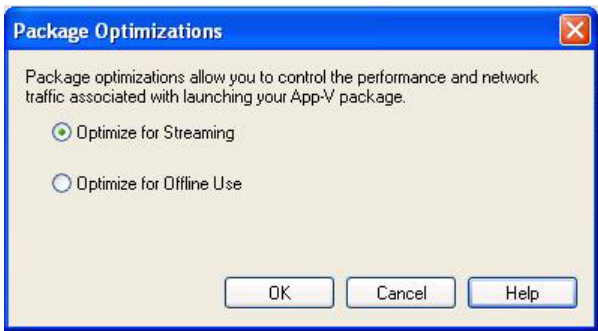




Figure 10-13: Package Optimizations Dialog Box

The **Package Optimizations** dialog box includes the following options:

Table 10-9 • Package Optimizations Options

Option	Description
Optimize for Streaming	<p>If you choose this option, the App-V Assistant will perform a static analysis of the shortcuts in the application and decide which files should be in feature block 1 and which should be in feature block 2.</p> <p>This option provides a relatively quick launch time while limiting network traffic during application use.</p>  <p>Note • When application files are streamed to a client either at launch or during application use, they are saved in the App-V cache and do not need to be streamed again during subsequent use of the application.</p>
Optimize for Offline Use	<p>If you choose this option, all files in the App-V application will be included in feature block 1 and will be streamed to the client at start up in one file before the application launches. After that, no more streaming is done. All files are stored in the App-V cache, which means that the application is available for use even when the machine is not connected to the App-V server.</p> <p>Select this option if you want to enable users to use the App-V application when not connected to the App-V server and if you want to eliminate network traffic when the App-V application is being used.</p>  <p>Note • When application files are streamed to a client either at launch or during application use, they are saved in the App-V cache and do not need to be streamed again during subsequent application use.</p>

Building an App-V Application

The method for building an App-V application depends upon what file you have open—an InstallShield project or a Windows Installer package.

- [Building an App-V Application for an InstallShield Project](#)
- [Building an App-V Application for a Windows Installer Package](#)

Building an App-V Application for an InstallShield Project

To build an App-V application for an InstallShield project, perform the following steps:



Task: *To build an App-V application for an InstallShield project:*

1. Open the InstallShield project in InstallShield.
2. On the **Releases** view of the Installation Designer, make sure that at least one release has been created, and select the release that you want to build.



Important • You cannot create or edit a release in the Microsoft App-V Assistant. If no releases exist, or if you want to create a new release, open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build an App-V application. For more information, see [Creating and Building Releases](#).

3. Open the **Build Options** page of the Microsoft App-V Assistant.
4. In the **Releases** tree, select the same release that is selected on the **Releases** view of the InstallShield Installation Designer. This is the release that you will build an App-V application for.
5. Click the **Build** toolbar button (or select **Build Release** on the **Build** menu) to start building the active release.



Important • When you select a release on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build an App-V application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what was selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and an App-V application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.

To build more than one release at a time, perform a batch build. See [Performing Batch Builds](#).

If you are building an App-V application for an InstallShield project, the output of the build is a Windows Installer package and an App-V application. For information on the files included in an App-V application, see [Components of an App-V Application](#).

Building an App-V Application for a Windows Installer Package

To build an App-V application for a Windows Installer package, perform the following steps:



Task:

To build an App-V application for a Windows Installer package:

1. Do one of the following to open a Windows Installer package:
 - On the **File** menu, select **Open** and select a Windows Installer package (.msi).
 - On the **File** menu, select **Open** and select a transform file (.mst). The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
 - On the **File** menu, select **New** to open the **New Project** dialog box. Select **Transform** and click **OK**. The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
2. Use the Installation Designer to make any desired edits to the Windows Installer package or Transform file, and use the Microsoft App-V Assistant to set App-V application options.
3. Save the edits to the Windows Installer package or transform file by selecting **Save** on the **File** menu.
4. On the **Build Options** page of the Microsoft App-V Assistant, select the **Build App-V application** option. The **Build Virtual Package** button is enabled.
5. Click the **Build Virtual Package** button to start building the App-V application.

If you are building an App-V application for a Windows Installer package, the output of the build will be an App-V application. For information on the files included in an App-V application, see [Components of an App-V Application](#).

Build Output

If you are building an App-V application for an InstallShield project, the output of the build is a Windows Installer package, an App-V application, and a new Basic MSI project. For information on the files included in an App-V application, see [Components of an App-V Application](#).

If you are building an App-V application for a Windows Installer package, the output of the build will be an App-V application and a new Basic MSI project. For information on the files included in an App-V application, see [Components of an App-V Application](#).

AdminStudio creates the new Basic MSI project in the release folder. Note the following details about this project:

- The project contains dynamic file links to the App-V application files.
- The properties and directories are updated in the project.
- The Releases view contains four or more releases that enable you to build different combinations of releases (for example, compressed or uncompressed, with or without an InstallShield prerequisite for the App-V client).

- A built release is also included with the new Basic MSI project. Any warnings or errors from this build are included as warning -9150 or error -9151 in the App-V project.
- You can use this project if you want to add more functionality to your Windows Installer package. For example, you can create major upgrades, digitally sign the package, and more.
- You can update the end-user license agreement (EULA) with your own custom EULA. The EULA that is included by default contains instructions that explain how to replace the default EULA text with your own EULA text.
- This project is overwritten each time that you rebuild your App-V project.



Note • For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see *Virtualization Conversion Errors and Warnings*.

Testing an App-V Application Using the App-V Application Launcher

You can use the InstallShield App-V Application Launcher to locally test a newly built App-V application before moving it to a deployment server.

To open the App-V Application Launcher, select **Test launch App-V application(s)** in the **More Options** list on the **Build Options** page. The App-V Application Launcher will attempt to launch that application. If there are multiple shortcuts (.osd files) in this App-V application, the **Launch App-V Application** dialog box opens, where you are prompted to select the shortcut you want to launch from a list of all of the shortcuts.

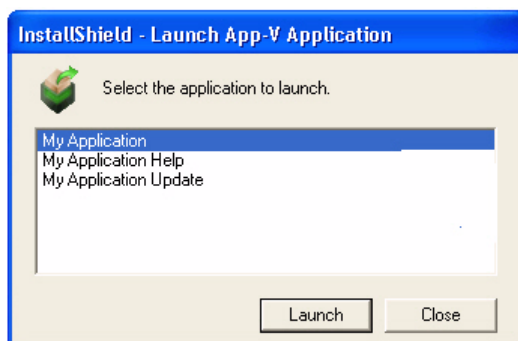
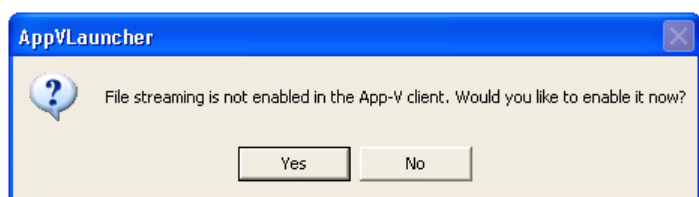


Figure 10-14: Launch App-V Application Dialog Box

Requirements for Using the App-V Application Launcher

There are two requirements for using the App-V Application Launcher:

- **Microsoft Virtualization client must be installed**—In order to test an App-V application using the App-V Application Launcher, you must have the Microsoft Virtualization Client installed on the machine.
- **File streaming must be enabled**—If you attempt to launch the App-V Application Launcher without having App-V file streaming enabled, an error message will open prompting you to enable the file streaming option. Click **Yes** to enable it.



App-V Application Launcher Utility Location

When an App-V application is built, the App-V Application Launcher utility (AppVLauncher.exe) is placed in the same folder as the App-V application (the same directory that contains the .sft and .osd files).

How the App-V Application Launcher Utility Works

On the **Package Information** page of the Microsoft App-V Assistant, you specify the URL of the App-V server where the .sft file of this App-V application will be deployed. This information is referenced in the .osd files in the generated App-V application. Therefore, testing an App-V application would seem to require that it be published on the server.

In order to make it possible to test the App-V package without having to publish it on the server or alter that URL path to the .sft file, a copy of the App-V Application Launcher utility is automatically copied to the output directory of each App-V application. Rather than looking for the .sft file in the location that you specified on the **Package Information** page, the App-V Application Launcher looks for the .sft file that is located in the same directory, ignoring the path set in the application.

The App-V Application Launcher is a convenient testing tool that makes it possible for you to reliably and accurately test your App-V applications on your local machine or any other system that has the App-V client installed before moving it to the App-V server.

Controlling Whether to Include the App-V Application Launcher in Output

If you do not want to include the App-V Application Launcher with each App-V application that you build, clear the **Create utility to test launch App-V applications as part of building App-V applications** check box on the **Virtual Packages** tab of the AdminStudio **Options** dialog box.

Microsoft App-V Assistant Reference

Reference information about the Microsoft App-V Assistant is organized into the following sections:

- [Pages](#)
- [Dialog Boxes](#)
- [Building App-V Applications Using the Command Line](#)
- [App-V Application Conversion Error and Warning Messages](#)
- [Application Features Requiring Pre- or Post-Conversion Actions](#)

Pages

The Microsoft App-V Assistant is comprised of the following pages:

- [Microsoft App-V Assistant Home Page](#)
- [Package Information Page](#)
- [Files Page](#)
- [Applications Page](#)
- [Registry Page](#)
- [Dynamic Suite Composition Page](#)
- [Build Options Page](#)

Microsoft App-V Assistant Home Page

The Microsoft App-V Assistant Home page displays a diagram that illustrates the process of creating an App-V application.

Click the following icons in the navigation bar at the bottom of the page to navigate through the Microsoft App-V Assistant interface:

Table 10-10 • Navigation Bar Icons








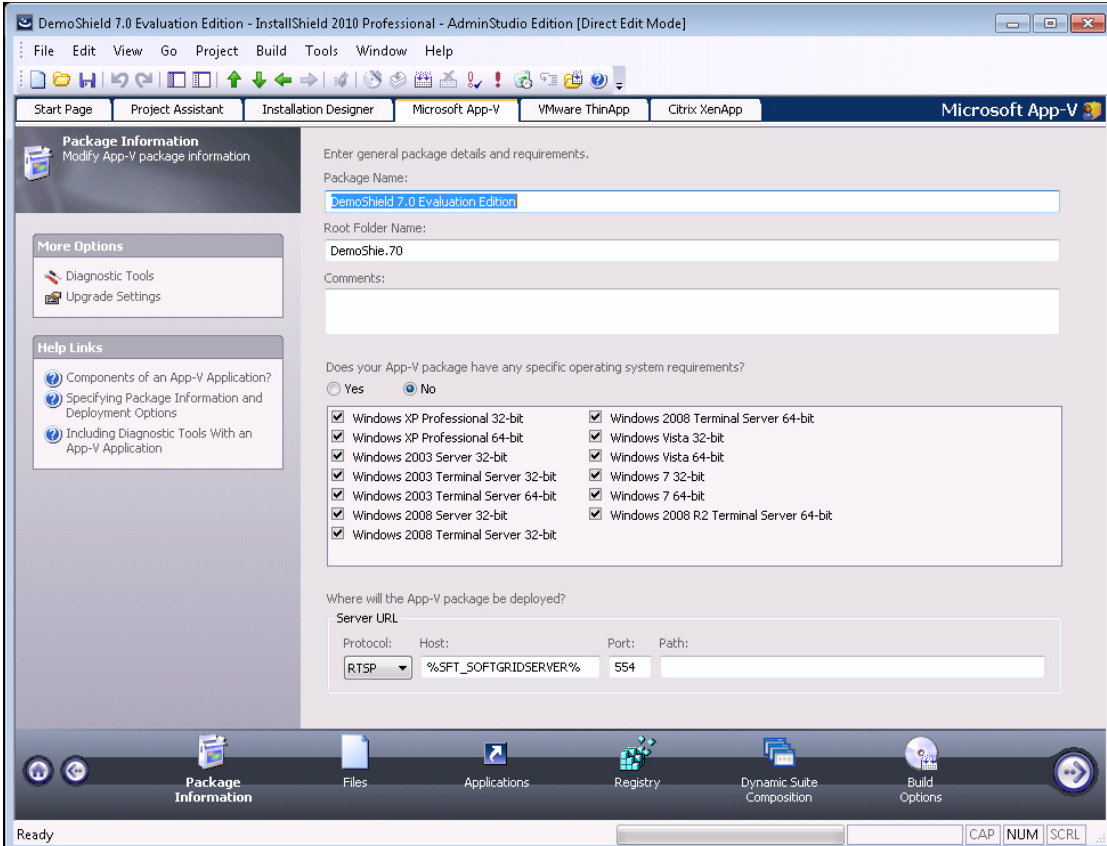
Icon	Destination
	Package Information Page
	Files Page
	Applications Page
	Registry Page
	Build Options Page
	Go to next page.

Table 10-10 • Navigation Bar Icons

Icon	Destination
	Jump back to previous page.
	Microsoft App-V Assistant Home Page

Package Information Page

On the Package Information page, you specify the package name and enter a comment to document this virtual package. From this page, you can also choose to include diagnostic tools with the virtual package.



DemoShield 7.0 Evaluation Edition - InstallShield 2010 Professional - AdminStudio Edition [Direct Edit Mode]

File Edit View Go Project Build Tools Window Help

Start Page Project Assistant Installation Designer Microsoft App-V VMware ThinApp Citrix XenApp Microsoft App-V

Package Information
Modify App-V package information

Enter general package details and requirements.

Package Name:
DemoShield 7.0 Evaluation Edition

Root Folder Name:
DemoShie.70

Comments:

More Options
Diagnostic Tools
Upgrade Settings

Help Links
Components of an App-V Application?
Specifying Package Information and Deployment Options
Including Diagnostic Tools With an App-V Application

Does your App-V package have any specific operating system requirements?
☐ Yes ☒ No

☒ Windows XP Professional 32-bit
☒ Windows XP Professional 64-bit
☒ Windows 2003 Server 32-bit
☒ Windows 2003 Terminal Server 32-bit
☒ Windows 2003 Terminal Server 64-bit
☒ Windows 2008 Server 32-bit
☒ Windows 2008 Terminal Server 32-bit
☒ Windows 2008 Terminal Server 64-bit
☒ Windows 2008 R2 Terminal Server 64-bit

Where will the App-V package be deployed?
Server URL
Protocol: Host: Port: Path:
RTSP %SFT_SOFTGRIDSERVER% 554

Package Information Files Applications Registry Dynamic Suite Composition Build Options

Ready CAP NUM SCRL

Figure 10-15: App-V Assistant Package Information Page

The Package Information page includes the following options:

Table 10-11 • Package Information Page



Option	Description
Package Name	<p>Enter a name (a maximum of 64 characters) to identify this virtual package.</p>  <p>Tip • In this field, you can specify the single name to use for an App-V application package that contains multiple applications. For example, Microsoft Office could be used to label a package that contains Microsoft Word and Microsoft Excel applications that run in the same virtual environment.</p>
Root Folder Name	<p>Enter a name to identify the directory that will contain the App-V application's .sft file. When an App-V application is configured on a client machine, it is mounted on the computer's Q:\ drive in the folder you specify in this setting.</p> <p>The default value for the Root Folder Name setting is based on the [ProductName] and [ProductVersion] properties of the App-V application's associated Windows Installer file using the 8.3 file naming convention. For example:</p> <ul style="list-style-type: none"> • If [ProductName] is MyApplication and [ProductVersion] is 1.12.3.1, the Default Folder Name will be MyApplic.112. • If [ProductName] is MyApp and [ProductVersion] is 1, then the Default Folder Name will be MyApp.1. • If [ProductName] is MyBlueApp and [ProductVersion] is 1.2.3.4, then the Default Folder Name will be MyBlueAp.123. <p>If you happen to have another App-V application that has identical values for the [ProductName] and [ProductVersion] properties, you can avoid conflicts by entering a unique value in the Root Folder Name setting.</p>
Comments	<p>Enter a short description of the purpose of this App-V application and any special information that you wish to document.</p>  <p>Note • The description you enter in this text box will appear in the OSD file ABSTRACT element.</p>

Table 10-11 • Package Information Page (cont.)

Option	Description
Does your App-V package have any specific operating system requirements?	<p>Select one of the following:</p> <ul style="list-style-type: none"> • Yes—Select this option if the application does not support one of the listed operating systems. When you select this option, the check boxes are unlocked and you can clear the selection of the unsupported operating systems. • No—Select this option if this application will run on all of the listed operating systems. When this option is selected, the operating system check boxes are locked and cannot be changed.
Operating System List	<p>If you set the Does your App-V package have any specific operating system requirements? option to Yes, this list becomes enabled.</p> <p>To specify operating system requirements, select the operating systems that this application supports, and clear those that this application does not support.</p>
Where will the App-V package be deployed?	<p>Optionally identify the Server URL where this App-V package will be deployed, as described in Specifying the Deployment Server.</p>
Diagnostic Tools	<p>For testing purposes, you can choose to include diagnostic tools in your App-V application by clicking the Diagnostic Tools link in the More Options area. For more information, see App-V Diagnostic Tools Dialog Box.</p>
Upgrade Settings	<p>To specify upgrade information for your App-V package, click the Upgrade Settings link in the More Options area. For more information, see App-V Package Upgrade Settings Dialog Box.</p>

Specifying the Deployment Server

To identify the server URL where this App-V package will be deployed, specify the following information:

Table 10-12 • Server URL Options



Option	Description
Protocol	<p>Enables you to select the protocol that will stream the sequenced application package from a virtual application server to an Application Virtualization Desktop Client. The following protocols are available:</p> <ul style="list-style-type: none"> • RTSP—The default, it specifies that the Real-Time Streaming Protocol controls the exchange of virtualization-enabled applications. • RTSPS—Specifies that the Real-Time Streaming Protocol with Transport Layer Security controls the exchange of a sequenced application package. • FILE—Specifies that the sequenced application will be streamed from a file share. • HTTP—Specifies that Hypertext Transport Protocol controls the exchange of a package. • HTTPS—Specifies that Secure Hypertext Transport Protocol controls the exchange of a package.
Host	<p>Specify the virtual application server or the load balancer in front of a group of virtual application servers that will stream the software package to an Application Virtualization Desktop Client.</p> <p>You must complete this item to create a sequenced application package, but you can change from the default %SFT_SOFTGRIDSERVER% environment variable to the actual hostname or IP address of a virtual application server.</p>  <p>Note • If you choose not to specify a static hostname or IP address, on each Application Virtualization Desktop Client you must set up an environment variable called <code>SFT_SOFTGRIDSERVER</code>.</p> <ul style="list-style-type: none"> • Its value must be the hostname or IP address of the virtual application server or load balancer that is this client's source of applications. • You should make this environment variable a system variable rather than a user variable. • Any Application Virtualization Desktop Client session that is running on this computer during your assignment of this variable must be closed and then opened so that the resumed session will be aware of its new application source.
Port	<p>Specify the port on which the virtual application server or the load balancer will listen for an Application Virtualization Desktop Client's request for the package.</p> <ul style="list-style-type: none"> • This information is required to create a package, but you can change it. • The default port is 554.

Table 10-12 • Server URL Options

Option	Description
Path	<p>Specify the relative path on the virtual application server where the software package is stored and from which it will be streamed.</p> <div></div> <hr/> <p>Note • <i>This information is required to create a package if the .sft file will be stored in a subdirectory of CONTENT; otherwise, this information is not required.</i></p>

Files Page

On the **Files** page of the Microsoft App-V Assistant, you can perform the following tasks:

- View Files and Folders
- Add Files and Folders
- Delete Files and Folders
- Set Isolation Options
- Modifying the Display of Predefined Folders

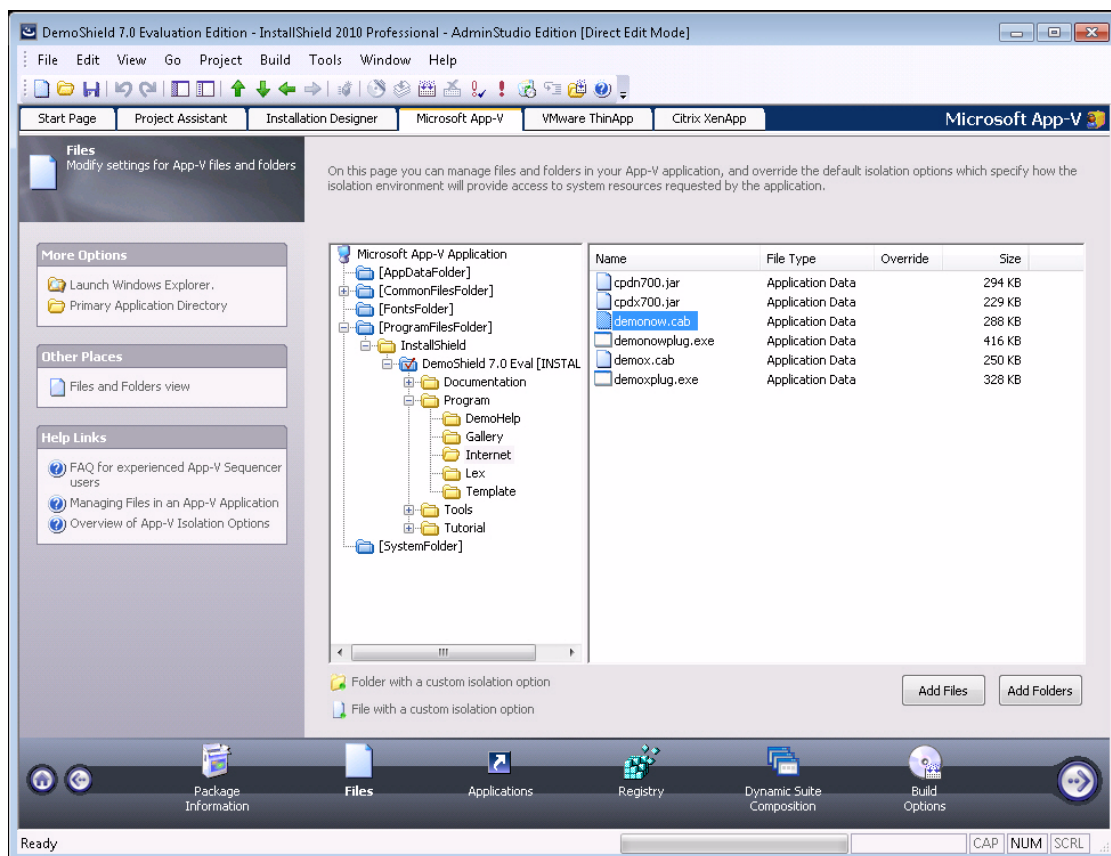


Figure 10-16: App-V Assistant Files Page

View Files and Folders

On the **Files** page, you can view all of the files and folders that are currently in your App-V application.

Folders are listed in the **Microsoft App-V application** tree on the left, and all of the files in the selected folder are listed on the right.

- The directories in the tree represent how your application will be organized within its secure compressed container.
- Blue folders are the supported MSI standard folders.
- The folder with the check mark is INSTALLDIR, which represents the main product installation directory.

Add Files and Folders

On the **Files** page, you can use the **Add Files** and **Add Folders** buttons to add new files and folders to include in the App-V application. See [Adding, Deleting, and Moving Files and Folders in an App-V Application](#).

If you are editing an InstallShield project (not a Windows Installer package), and you are adding a folder to this App-V application, you are prompted to choose whether you want to create a dynamic file link to the source folder.

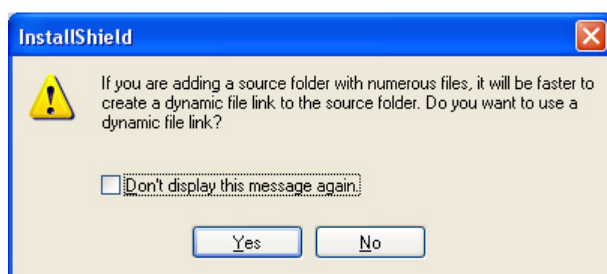


Figure 10-17: Dynamic File Link Option Dialog Box

Indicate whether you want to create a dynamic file link by selecting one of the following:

- **No**—For more flexibility with App-V options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
- **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See Dynamic File Link Settings Dialog Box.

Delete Files and Folders

You can delete files and folders from the App-V application by selecting the file or folder you want to delete, and selecting **Delete** from the context menu. For more information, see [Deleting Files and Folders](#).



Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Set Isolation Options

When using the Microsoft App-V Assistant, you can set isolation options to control application compatibility and accessibility. The isolation option that is assigned to a file or folder specifies how the isolation environment will provide access to system resources requested by the application.

You set isolation options on the **Isolation Options** dialog box, which is opened by selecting **Isolation Options** on the context menu when you have a file or folder selected on the **Files** page or a registry key selected on the **Registry** page.

You set isolation options by selecting a file or folder and then selecting **Isolation Options** from the context menu. For an overview of the available isolation options, and for instructions on how to set them, see [Setting Isolation Options for Folders and Files](#).

Modifying the Display of Predefined Folders

You can specify which of the Windows Installer predefined folders are listed in the **Microsoft App-V application** tree. See [Controlling the Display of Predefined Folders](#).

Applications Page

You define shortcuts to enable users to launch an App-V application from within the virtual environment.

By default, the **Microsoft App-V Assistant** creates App-V applications for all of the executable shortcuts that exist in your project. The project's shortcuts are listed in a checklist on the **Applications** page.

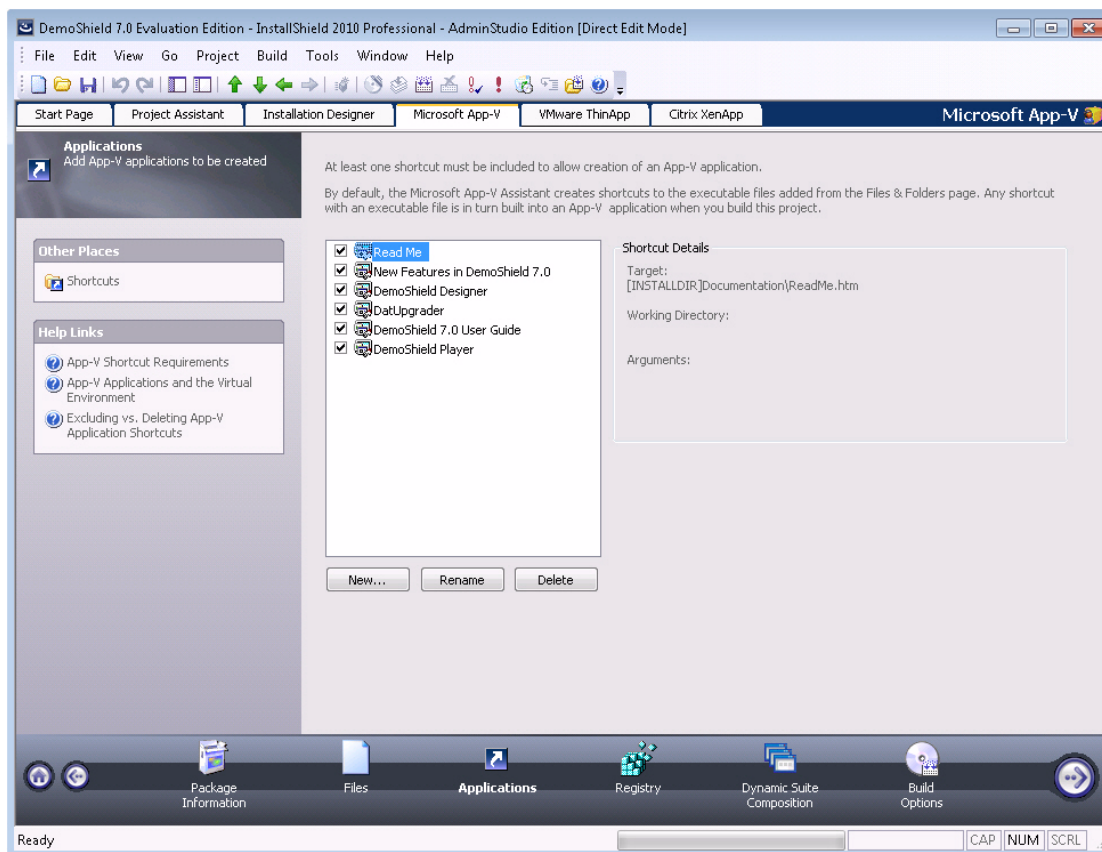


Figure 10-18: App-V Assistant Applications Page

Shortcut Requirements

For each App-V application, you are required to define **at least one** shortcut. You define application shortcuts to enable users to launch an App-V application from within the virtual environment. By default, the Microsoft App-V Assistant creates App-V applications for all of the executable shortcuts that exist in your project (or Windows Installer package).

If you build an App-V application that does not contain any shortcuts, users will not be able to launch the application.

Difference Between Deleting and Excluding a Shortcut

To prevent a shortcut from being created in the App-V application, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the App-V application, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding an App-V Application](#).
- **Deleting a Shortcut**—When you delete a shortcut, it is removed from both the App-V application and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting an App-V Application](#).

Managing Shortcuts

On the **Applications** page, you can create, delete, include, exclude, or rename an App-V application. For step-by-step instructions, see the following topics:

- [Creating a New App-V Application](#)
- [Including an Existing App-V Application](#)
- [Excluding or Deleting an Existing App-V Application](#)
- [Renaming an App-V Application](#)

Registry Page

On the **Registry** page, you can view existing registry keys, values, and data, and add or delete registry items. You can also override the default isolation options for a registry key. Isolation options specify how the virtual environment will provide access to system resources requested by the application.

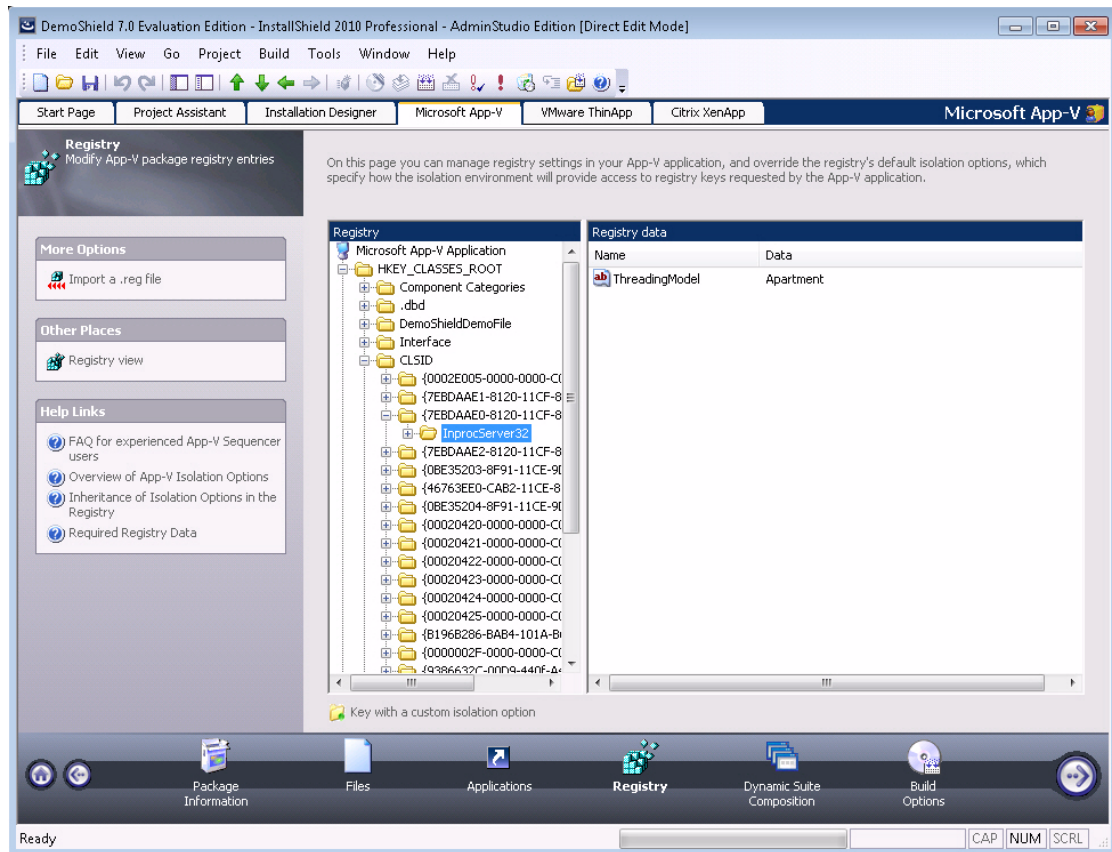


Figure 10-19: App-V Assistant Registry Page

The default settings for isolation options are built into the Microsoft App-V Assistant, and those defaults are adequate for most environments. However, you can override the default settings for selected registry keys to exert control over application interactions with client operating system resources.

You set isolation options by selecting a registry key and then selecting **Isolation Options** from the context menu. For an overview of the available isolation options, and for instructions on how to set them, see [Setting Isolation Options for Folders and Files](#).

Registry items that are listed on this page will be included in the App-V application, and those that you delete will not. By default, all new registry keys are isolated.



Tip • To import an existing registry (.reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.



Note • You cannot set isolation options on root registry keys.

Editing the registry on the **Registry** page is performed much like it is performed on the InstallShield **Registry View**. See [Editing the Registry](#).



Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

Dynamic Suite Composition Page

The point of application virtualization is to minimize the system dependencies that an application has on the underlying physical system. Many applications have common system dependencies on plug-ins or middleware, such as Adobe Reader or ODBC drivers.

Dynamic Suite Composition (DSC) is a Microsoft Application Virtualization feature that enables applications to be virtualized separately from the plug-ins and middleware applications that they rely on, while still enabling them to communicate with those plug-ins and middleware applications within the virtual environment. The primary App-V application and the dependency App-V applications in the dynamic suite will run and interact with one another as if they were all installed locally on a computer. You would only need to deploy common system components once on each client, making them available for use by many App-V applications, rather than to include them with each of the App-V applications that are dependent upon them. This reduces redundancy in the local App-V cache and simplifies the construction and testing of the primary App-V applications.

You specify the App-V applications that you want to include in a dynamic suite on the **Dynamic Suite Composition** page of the App-V Assistant.

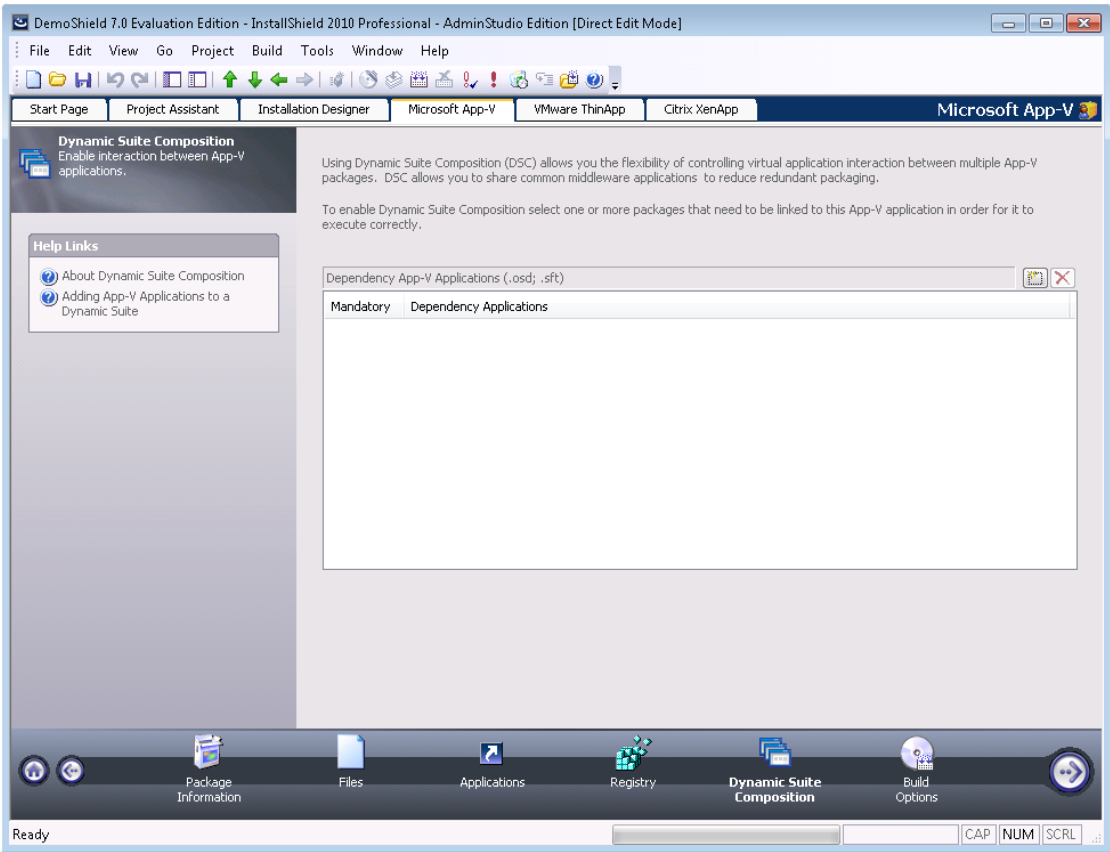




Figure 10-20: Microsoft App-V Assistant Dynamic Suite Composition Page

The **Dynamic Suite Composition** page has the following options:

Table 10-13 • Dynamic Suite Composition Page

Option	Description
Dependency App-V Applications List	List of all of the dependency App-V applications that have been selected for the primary App-V application.
Mandatory	Indicates whether to selected dependency App-V application is required in order for the primary App-V application to run. <ul style="list-style-type: none">• If the dependency App-V application is mandatory—If the primary App-V application will not run unless it can locate this dependency App-V application, leave the Mandatory option selected. If a dependency App-V application that is configured as Mandatory is not available, an error will be generated when someone attempts to run the primary App-V application.• If the dependency App-V application is not mandatory—If the primary App-V application will run even if it cannot locate this dependency App-V application, clear the Mandatory option.

Table 10-13 • Dynamic Suite Composition Page

Option	Description
New Button 	<p>To add an App-V application to the Dependency App-V Applications list, click this button and select the App-V application (.osd, .sft) that you want to add. Select one of the following:</p> <ul style="list-style-type: none"> One of the .osd files—If this dependency App-V application is or is going to be published on a server, select any one of the .osd files that are listed. If these .osd files were created properly, each of them should contain the information that will identify to the primary App-V application the published location of the dependency App-V application. It is not necessary to select more than one .osd file. All of them contain the same reference to the location of the dependency App-V application's .sft file, which is the only reference that is necessary in order for the primary App-V application to locate it. The .sft file—If this dependency App-V application is or is going to be published locally on the client or at an accessible network location, you may select just the .sft file. <p>The selected reference App-V application is now listed in the Dependency Applications list and, by default, the Mandatory option is selected.</p>
Delete Button 	Click to delete the selected App-V application from the list.

Build Options Page

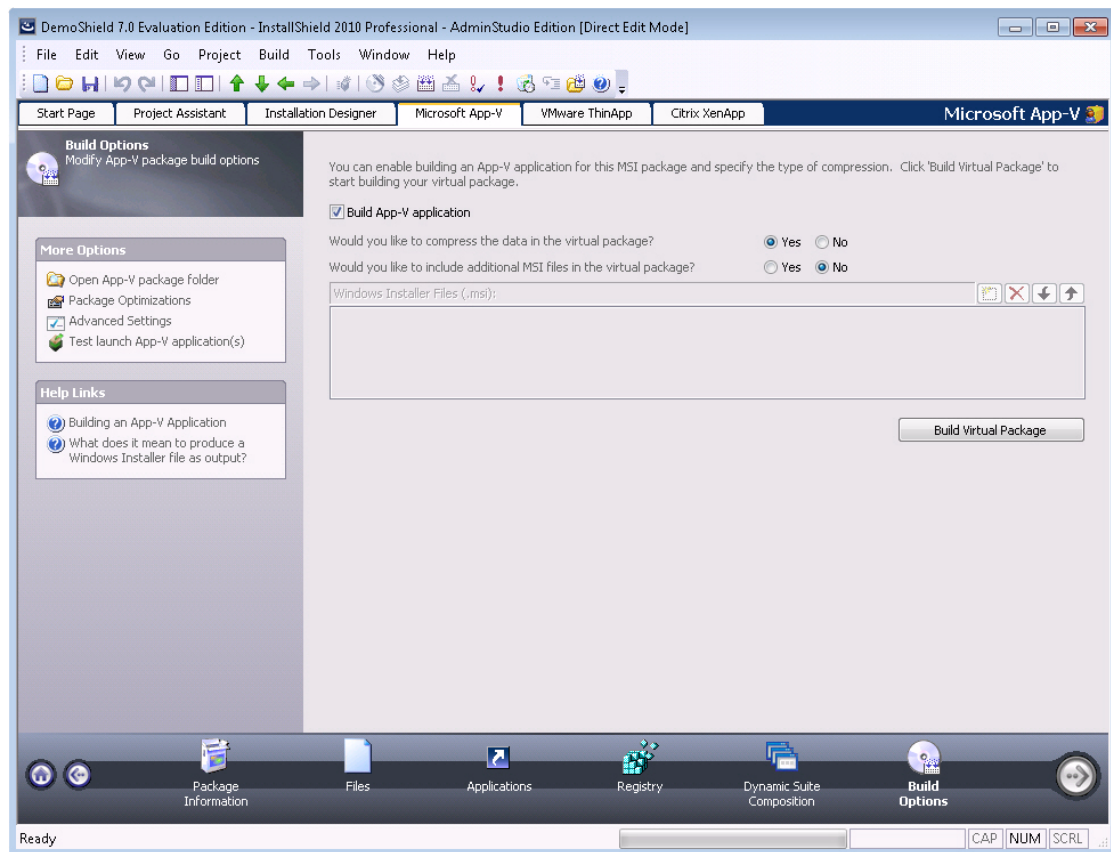


Figure 10-21: App-V Assistant Build Options Page

The options on the Build Options page vary depending upon whether you are editing an InstallShield project or a Windows Installer package:

- **InstallShield Project [Basic MSI Project Mode]**—When you open an InstallShield project in InstallShield, the following conditions are true:
 - The **Build Options** page includes a releases tree, and you select the release that you want to build.
 - To build the App-V application, you click the **Build** button on the toolbar.
- **Windows Installer Package [Direct Edit Mode]**—When you open a Windows Installer package in InstallShield, the following conditions are true:
 - Because you do not have to select a release for a Windows Installer package, there is no releases tree.
 - Because a Windows Installer package has already been built, InstallShield's standard build functionality is disabled. To build the App-V application, select the **Build App-V application** option and click the **Build Virtual Package** button.

Specifying Build Options

On the **Build Options** page, you can specify the following options:




Table 10-14 • App-V Application Build Options


Option	Description
Build App-V application	<p>(Direct Edit/Direct MST Modes Only) When you directly edit a Windows Installer package, it is not necessary to build the package, because it is already built. Therefore, InstallShield's Build function is disabled. Select the Build App-V application option to enable the Build function. When this option is selected, the Build Virtual Package button is enabled.</p> <p>For more information, see Enabling App-V Application Building When in Direct Edit Mode.</p>
Build Virtual Package	<p>(Direct Edit/Direct MST Modes Only) When you directly edit a Windows Installer package, if you select the Build App-V application option, this button is enabled. Click it to build the App-V application.</p>
Would you like to compress the data in the virtual package?	<p>Set this option to Yes to use zlib compression to compress the data in the App-V package</p>
Would you like to include additional MSI files in the virtual package?	<p>Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. To include additional Windows Installer packages in an App-V application, set this option to Yes, and then select the packages that you want to add.</p> <div data-bbox="561 1215 596 1262" data-label="Image"> </div> <p>Note • For more information, see Including Additional Windows Installer Packages in an App-V Application.</p>

Including Additional Windows Installer Packages in an App-V Application

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual package, you can also use the Microsoft App-V Assistant to convert an application suite of multiple Windows Installer packages into one virtual package.

To include additional Windows Installer packages in an App-V application, set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**, and then select the packages that you want to add.

- Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.
- The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.

- Use the Delete button () to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Selecting Releases to Build

You select the releases that you want to build an App-V application for on the **Releases** tree of the **Build Options** page. By selecting a release, you are specifying that whenever that particular release is built, an App-V application will also be built.



Note • If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.

About Building Releases

When you select a release on the Releases tree on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build an App-V application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and an App-V application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See [Performing Batch Builds](#).

About Creating Releases

You cannot create or edit a release in the Microsoft App-V Assistant. If no releases exist, you can simply click the **Build** toolbar button to create a new release or open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build an App-V application. For more information, see [Creating and Building Releases](#).

If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.

Choosing to Build a Windows Installer Package to Assist in the Distribution of an App-V Application

You can choose to build a Windows Installer package to assist in the distribution of an App-V application by selecting the **Generate an installation package as part of the build output** check box on the [Advanced Settings Dialog Box](#). You can access this dialog box by clicking the **Advanced Settings** link in the **More Options** area on the **Build Options** page of the Microsoft App-V Assistant. By default, this check box is not selected.

When you run this Windows Installer file, the minimally required App-V application files will be “installed” in the local App-V client system cache. (The .sft file remains on the App-V server until the client requests that it be downloaded when the application is launched for the first time.)



Note • The App-V client must be installed on the local machine before you can install an App-V application. The installation will detect and warn if the App-V client is not available, and the installation will fail.

This simplifies the deployment of an App-V application by enabling you to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management.

To remove an installed App-V application, you need to use the Application Virtualization Client tool, which is available in the Administrative Tools of the Windows Control Panel.

Enabling App-V Application Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **Microsoft App-V Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It is not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build an App-V application for this Windows Installer package. To enable the **Build** button to build just the App-V application, select the **Build App-V application** option on the **Build Options** page.

After you select this option, the **Build App-V application** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.

Specifying Package Optimizations

You can use the App-V Assistant **Package Optimizations** feature to control the performance and network traffic associated with running an App-V application. The package optimization option you select determines how quickly the App-V application will launch, and how often additional functionality will need to be streamed to the client while the App-V application is being used.

You indicate your package optimization preference on the [Package Optimizations Dialog Box](#), which is opened by clicking the **Package Optimizations** link in the **More Options** menu of the **Build Options** page.

Building an App-V Application

The method for building an App-V application depends upon what file you have open—an InstallShield project or a Windows Installer package. For detailed instructions, see one of the following topics:

- [Building an App-V Application for an InstallShield Project](#)
- [Building an App-V Application for a Windows Installer Package](#)

Testing an App-V Application Using the App-V Application Launcher

From the **Build Options** page, you can choose to test your App-V application using the App-V Application Launcher. To open the App-V Application Launcher, select **Test launch App-V application(s)** in the **More Options** list on the **Build Options** page.

For more information, see [Testing an App-V Application Using the App-V Application Launcher](#) and [Launch App-V Application Dialog Box](#).

Opening the App-V Application Folder

To quickly open the folder containing the App-V application files that were generated when this InstallShield project or Windows Installer package was built, click **Open App-V application folder** in the **More Options** list.

Dialog Boxes

The Microsoft App-V Assistant includes the following dialog boxes:

- [App-V Diagnostic Tools Dialog Box](#)
- [File Isolation Options Dialog Box](#)
- [Folder Isolation Options Dialog Box](#)
- [Launch App-V Application Dialog Box](#)
- [Package Optimizations Dialog Box](#)
- [Primary Application Directory Dialog Box](#)
- [Registry Isolation Options Dialog Box](#)


Advanced Settings Dialog Box



Edition • The Microsoft App-V Assistant is included in the Virtualization Pack.

The Advanced Settings dialog box opens when you click the Advanced Settings link in the More Options section on the Build Options tab. This dialog box is where you specify build and run-time options.

Table 10-15 • Advanced Settings Dialog Box Options

Option	Description
Generate an installation package as part of the build output	<p>To build an installation package with your App-V application, select this check box.</p> <p>This check box is not selected by default. If you do select this check box, you can specify whether you want to load the installation package from the media location or from the shared location.</p> <p>Building an installation package enables you or your end users to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management to distribute your App-V application.</p> <p>When you run this Windows Installer file, the minimally required App-V application files will be “installed” in the local App-V client system cache. (The .sft file remains on the App-V server until the client requests that it be downloaded when the application is launched for the first time.)</p>  <p>Note • The App-V client must be installed on the local machine before you can install an App-V application. The installation will detect and warn if the App-V client is not available, and the installation will fail.</p> <p>To remove an installed App-V application, you need to use the Application Virtualization Client tool, which is available in the Administrative Tools of the Windows Control Panel.</p>
Load from Media Location	<p>To load the installation package from the media location, select this option. If you select this option, the following check boxes are available for selection:</p> <ul style="list-style-type: none"> • Compress—To build a compressed installation package, select this check box. If this check box is cleared, an uncompressed installation package is built. • App-V Client Prerequisite (Generates Setup.exe)—If you want to include the AdminStudio prerequisite that installs the App-V client on the target system, select this check box. Note that a Setup.exe setup launcher is required if the AdminStudio prerequisite needs to be included in the release.
Load from Shared Location	<p>To load the installation package from the shared location, select this check box.</p>
App-V application Test Launcher	<p>To use the App-V application test launcher to locally test a newly built App-V application before moving it to a deployment server, select this check box. For more information, see Testing an App-V Application Using the App-V Application Launcher.</p>

App-V Diagnostic Tools Dialog Box

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **Package Information** page, you can choose to include the Registry Editor and Windows Command Prompt diagnostic tools with your App-V application.

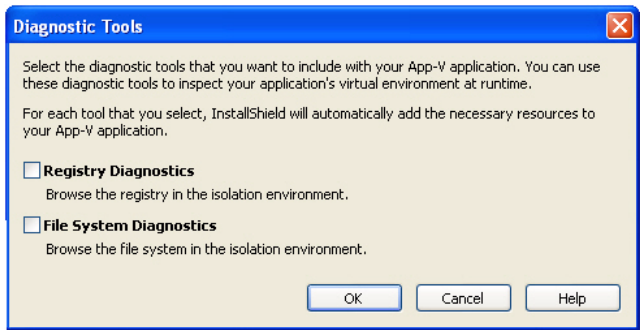


Figure 10-22: Diagnostic Tools Dialog Box

If you include diagnostic tools with your App-V application, you will be able to look at the registry or file system for the application while it is running in its virtual environment. For example, if you were running an App-V application and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.



Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the App-V application. However, these tools may not be compatible with other operating systems.

You can use these diagnostic tools to inspect your application's virtual environment at runtime. You have the following options:

Table 10-16 • Diagnostic Tools Dialog Box Options

Option	Description
Registry Diagnostics	<p>Select this option if you want to include the Registry Editor (<code>regedit.exe</code>) with your App-V application so that you can browse the registry at runtime from within the virtual environment.</p> <p>If this option is selected, a file named <code>Virtual Registry.osd</code> will be created in the App-V Package folder, which can be used to display the registry within the virtual environment. You can use <code>Virtual Registry.osd</code> to view the existing registry on the computer plus the registry for the virtual package.</p>

Table 10-16 • Diagnostic Tools Dialog Box Options

Option	Description
File System Diagnostics	<p>Select this option if you want to include the Windows Command Prompt application with your App-V application so that you can browse the virtual file system at runtime from within the virtual environment.</p> <p>If this option is selected, a file named <code>Virtual File System.osd</code> will be created in the App-V Package folder, which can be used to display the files and folders within the virtual environment. You can use <code>Virtual File System.osd</code> to view the existing files and folders on the computer plus the files and folders for the virtual package.</p>

Launching the Diagnostic Tools Within the Virtual Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the App-V application.

When the user runs this App-V application, two additional shortcuts will be available in the application's shortcut folder: The names of these shortcuts will reflect the application name, such as:

```
[ProductName] Registry
[ProductName] File System
```

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application's virtual environment.

App-V Package Upgrade Settings Dialog Box

The App-V Package Upgrade Settings dialog box is where you specify whether you want to create an upgrade for your App-V package. If you specify that you do want to create an upgrade, you can specify additional information about the upgrade.

Table 10-17 • App-V Package Upgrade Settings Dialog Box Settings

Setting	Description
Enable Upgrade	<p>To create an upgrade for an earlier App-V package, select this check box.</p> <p>If you do not want to create an upgrade, clear this check box. When this check box is cleared, the other settings on the App-V Package Upgrade Settings dialog box are disabled.</p> <p>This check box is cleared by default.</p>
Always upgrade latest built package	<p>If you selected the Enable Upgrade check box and you want InstallShield to build an upgrade that updates the latest built App-V package, select this option.</p>

Table 10-17 • App-V Package Upgrade Settings Dialog Box Settings (cont.)

Setting	Description
Choose package to upgrade	If you selected the Enable Upgrade check box and you want InstallShield to build an upgrade that updates a particular App-V package, select this option, and then specify the path of the earlier package that you want to be updated.
Append version number to package name	If you selected the Enable Upgrade check box and you want InstallShield to append the version number to the App-V package name, select this check box.

File Isolation Options Dialog Box

On the **File Isolation Options** dialog box, you can override the default isolation options for the selected file.



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, App-V, and registry settings.

The **File Isolation Options** dialog box includes the following options:

Table 10-18 • File Isolation Options Dialog Box

Option	Description
File Type	<p>When an App-V application performs a WRITE operation to a resource (a file or folder), the resource's File Type setting determines whether changes to that resource are saved for all users of the App-V application on that machine or only for the logged-in user. Select one of the following options:</p> <ul style="list-style-type: none"> • Application Data—Changes to the resource are saved for all users of this App-V application on this machine. • User Data—Changes to the resource are saved only for the logged-in user.
Override	<p>To have the virtualization client overwrite the file when the application is upgraded and streamed from the App-V server to the client, select this check box.</p> <p>If Override check box is not selected, then the App-V client determines whether to overwrite the file during an upgrade. In general, if the file is marked as Application Data, the file is overwritten during an upgrade; otherwise, it is not overwritten.</p>

Folder Isolation Options Dialog Box

On the **Folder Isolation Options** dialog box, you can override the default isolation options for the selected folder.



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, App-V, and registry settings.*

The **Folder Isolation Options** dialog box includes the following options:

Table 10-19 • Folder Isolation Options Dialog Box

Option	Description
File Type	<p>When an App-V application performs a WRITE operation to a resource (a file or folder), the resource's File Type setting determines whether changes to that resource are saved for all users of the App-V application on that machine or only for the logged-in user. Select one of the following options:</p> <ul style="list-style-type: none">• Application Data—Changes to the resource are saved for all users of this App-V application on this machine.• User Data—Changes to the resource are saved only for the logged-in user.
Override	<p>To have the virtualization client overwrite the folder when the application is upgraded and streamed from the App-V server to the client, select this check box.</p> <p>If Override check box is not selected, then the App-V client determines whether to overwrite the folder during an upgrade. In general, if the folder is marked as Application Data, the file is overwritten during an upgrade; otherwise, it is not overwritten.</p>

Launch App-V Application Dialog Box

When using the InstallShield App-V Application Launcher to locally test a newly built App-V application before moving it to a deployment server, the **Launch App-V Application** dialog box opens when an App-V application has more than one shortcut (.osd files). You are prompted to select the shortcut you want to launch from a list of all of the shortcuts.

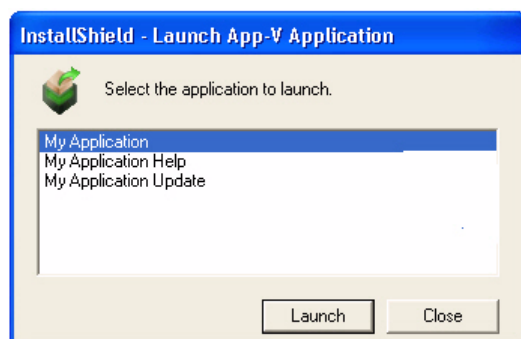


Figure 10-23: Launch App-V Application Dialog Box

To open the App-V Application Launcher, select **Test launch App-V application(s)** in the **More Options** list on the **Build Options** page.

Package Optimizations Dialog Box

You can use the App-V Assistant **Package Optimizations** feature to control the performance and network traffic associated with running an App-V application. The package optimization option you select determines how quickly the App-V application will launch, and how often additional functionality will need to be streamed to the client while the App-V application is being used.

You indicate your package optimization preference on the **Package Optimizations** dialog box, which is opened by clicking the **Package Optimizations** link in the **More Options** menu of the **Build Options** page.

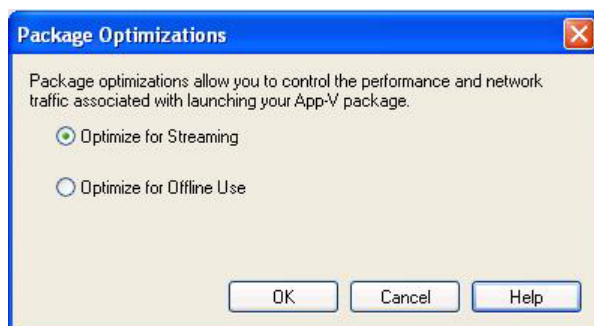


Figure 10-24: Package Optimizations Dialog Box



The files in an App-V application can be grouped into two feature blocks:

- **Feature block 1**—Feature block 1 must contain the core functionality of the App-V application that is necessary to launch the application. At application launch, all of the files in feature block 1 are streamed to the client in one unit.
- **Feature block 2**—Feature block 2 can contain additional functionality of the App-V application that is not necessary to launch the application. While the App-V application is being used, the files in feature block 2 can be streamed in small packets on an as-needed basis.

You can either choose to include all App-V application files in feature block 1 or, to improve launch speed, you can choose to group the files into two feature blocks: feature block 1 and feature block 2.

The **Package Optimizations** dialog box includes the following options:

Table 10-20 • Package Optimizations Options

Option	Description
Optimize for Streaming	<p>If you choose this option, the App-V Assistant will perform a static analysis of the shortcuts in the application and decide which files should be in feature block 1 and which should be in feature block 2.</p> <p>This option provides a relatively quick launch time while limiting network traffic during application use.</p>  <p>Note • When application files are streamed to a client either at launch or during application use, they are saved in the App-V cache and do not need to be streamed again during subsequent use of the application.</p>
Optimize for Offline Use	<p>If you choose this option, all files in the App-V application will be included in feature block 1 and will be streamed to the client at start up in one file before the application launches. After that, no more streaming is done. All files are stored in the App-V cache, which means that the application is available for use even when the machine is not connected to the App-V server.</p> <p>Select this option if you want to enable users to use the App-V application when not connected to the App-V server and if you want to eliminate network traffic when the App-V application is being used.</p>  <p>Note • When application files are streamed to a client either at launch or during application use, they are saved in the App-V cache and do not need to be streamed again during subsequent application use.</p>

Primary Application Directory Dialog Box

When App-V applications are run on a machine that has Microsoft Virtualization Client installed, they are run from the Client's application virtual drive.

For optimum performance, it is desirable to have the bulk of the application's files mounted to this drive at run-time. In order to achieve this, it is useful to determine an application's primary application directory so that folder can be mounted to the App-V Client's application virtual drive when the App-V application is loaded.

The InstallShield Microsoft App-V Assistant uses a series of steps to determine an App-V application's default primary application directory (as described in [Determining the Default Primary Application Directory](#)).

However, you can specify the primary application directory for an App-V application on the **Primary Application Directory** dialog box, which is opened by clicking **Primary Application Directory** in the **More Options** list on the **Files** page.

The **Primary Application Directory** dialog box displays the current primary application directory setting (if one has already been specified).

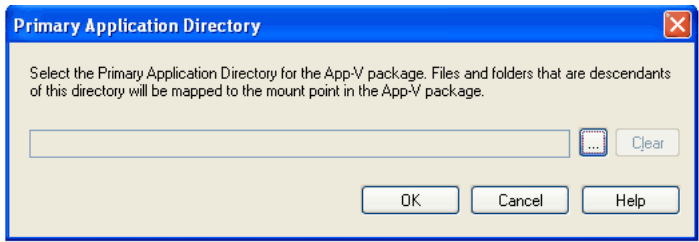


Figure 10-25: Primary Application Directory Dialog Box

When you click the browse (..) button, the **Browse for Directory** dialog box opens, listing all of the currently available destination directories for this App-V application. Select one of the listed directories and click **OK**.



Determining the Default Primary Application Directory

When an App-V application is built using InstallShield or any AdminStudio tool, the following series of steps are used to determine an App-V application's primary application directory:

Table 10-21 • Steps to Automatically Determine Primary Application Directory

#	Step	Description
1	Explicitly set primary application directory	If a directory is specified on the InstallShield App-V Assistant's Primary Application Directory dialog box (as described in Explicitly Setting the Primary Application Directory), that directory will be used.

Table 10-21 • Steps to Automatically Determine Primary Application Directory

#	Step	Description
2	Value of INSTALLDIR variable	<p>If the Windows Installer package has a value for INSTALLDIR (a system variable that specifies the root destination directory for an installation), that value will be used as the primary application directory.</p>  <p>Note • This step is not used when converting Windows Installer packages that have been repackaged using AdminStudio Repackager.</p>  <p>Note • All Windows Installer packages created by InstallShield or AdminStudio will have a value for the INSTALLDIR variable.</p>
3	Location of shortcut in a subdirectory of the ProgramFilesFolder	<p>If one of the .exe targets for a shortcut is in a subdirectory of ProgramFilesFolder, then the folder directly under ProgramFilesFolder will be used as the primary application directory. Typically, this would be:</p> <p>C:\Program Files\YourApplication</p>
4	Location of shortcut in a directory other than ProgramFilesFolder	<p>If no .exe targets are located in a subdirectory of ProgramFilesFolder, then the target directory of a shortcut that contains an .exe will be used.</p>
5	ProgramFilesFolder	<p>If none of the above can be found, then the primary application directory will be set to ProgramFilesFolder. Typically, this would be:</p> <p>C:\Program Files</p>

Registry Isolation Options Dialog Box

You open the **Registry Isolation Options** dialog box by selecting a registry key on the **Registry** page and clicking **Isolation Options** on the context menu. On the **Registry Isolation Options** dialog box, you can specify whether the App-V application will see only the registry entries for the selected key that are part of that App-V application, or will see a merged view of the registry entries for the selected key from both the local registry and from the App-V application's registry.

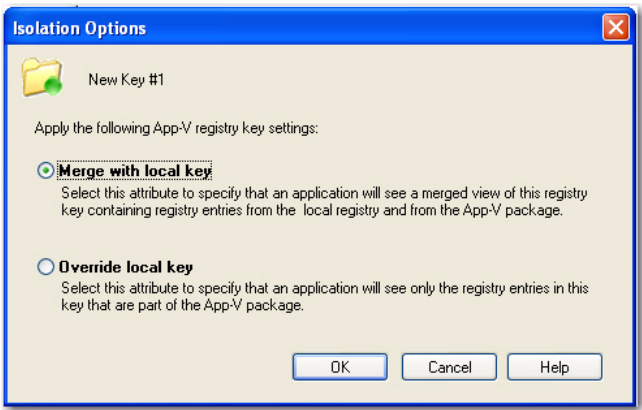


Figure 10-26: Registry Isolation Options Dialog Box



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.

The **Registry Isolation Options** dialog box includes the following options;

Table 10-22 • Registry Isolation Options

Option	Description
Merge with local key	Select this option to specify that the App-V application will see a merged view of the registry entries for the selected key from both the local registry and from the App-V application's registry.
Override local key	Select this option to specify that the App-V application will see only the registry entries for the selected key that are part of that App-V application.

Building App-V Applications Using the Command Line

When you configure an App-V application in an InstallShield project and then build that project (using either the user interface or the command line), both the Windows Installer package and the App-V application are built. When you use the standard InstallShield command line build, you do not need to add any additional command line parameters. All of the App-V application settings are saved within the InstallShield project.

App-V Application Conversion Error and Warning Messages



Note • For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see *Virtualization Conversion Errors and Warnings*.

Application Features Requiring Pre- or Post-Conversion Actions

Some application features are ignored when creating an App-V application. Therefore, some additional pre- or post-conversion actions must be taken in order for the App-V application to be created properly.

One action you could take to try to include ignored features in an App-V application is to first repackage the application using the Repackaging Wizard, and then convert the repackaged application to an App-V application.

For a list of ignored features, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

Creating ThinApp Applications



Edition • The ThinApp Assistant is included in the Virtualization Pack.



Important • ThinApp support requires a separate purchase of VMware® ThinApp™.

ThinApp (formerly Thinstall Virtualization Suite) is a self-contained application virtualization solution that requires no client-side agents or supporting server infrastructure. A ThinApp application runs within a virtual environment that prevents it from interfering with other software running on the same machine.

You can use the InstallShield **ThinApp Assistant** to configure and build a ThinApp application. Information about creating ThinApp applications using the ThinApp Assistant is presented in the following sections:

- [Overview of the ThinApp Assistant](#)
- [Using the ThinApp Assistant to Create a ThinApp Application](#)
- [ThinApp Assistant Reference](#)
- [ThinApp Application Configuration File: package.ini](#)



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

Overview of the ThinApp Assistant

A ThinApp application is a self-contained application virtualization solution that requires no client-side agents or supporting server infrastructure. A ThinApp application runs within a virtual environment that prevents it from interfering with other software running on the same machine. You can use the InstallShield **ThinApp Assistant** to configure and build a ThinApp application.

- [About ThinApp Applications](#)
 - [The ThinApp Virtual Operating System](#)
 - [Benefits of Deploying ThinApp Applications](#)
- [About the ThinApp Assistant](#)
 - [Process for Authoring a ThinApp Application Using the ThinApp Assistant](#)
 - [Components of a ThinApp Application](#)
 - [Supported InstallShield Project Types](#)
 - [How Transforms are Included in a ThinApp Application](#)
 - [About Sandboxes](#)

About ThinApp Applications

ThinApp applications can be deployed on a machine without modifying the local operating system or file system. They run in a “sandbox” (or virtual environment) which protects the local operating system from installation modifications that could affect stability or security. Also, ThinApp applications can be run safely from restricted user accounts without local installation.

Information about ThinApp applications is presented in the following sections:

- [The ThinApp Virtual Operating System](#)
- [Benefits of Deploying ThinApp Applications](#)

The ThinApp Virtual Operating System

A ThinApp application runs in a virtual operating system—a small light-weight component which is embedded with each ThinApp application—that consists of a virtual file system and a virtual registry. When the ThinApp application is run, the virtual operating system environment is merged with the real system environment.

The virtual operating system technology enables entire applications to be packaged into a single .exe file that can be run without an installation process, and without modifying the resident operating system.

A ThinApp application can be run from a network or offline on the local machine.

Benefits of Deploying ThinApp Applications

Deploying ThinApp applications provides the following benefits:

- **Reduces time to deployment and costs associated with testing**—Applications can be deployed and run in independent sandboxes, eliminating the need for expensive and time-consuming multi-application regression testing. This reduces the time to deployment and the costs associated with testing.
- **Fast, lightweight virtualization**—ThinApp does not use emulation, so all processes are executed natively at full speed.
- **Reduces the cost of maintaining secure locked-down desktops**—ThinApp applications can run in restricted user accounts without requiring any host modifications.
- **Enhances work-force mobility, business continuity and disaster recovery**—ThinApp applications can be run off-line, directly from any external media including USB Flash, CD-ROM, and off-line laptops.
- **No infrastructure changes needed**—ThinApp applications can be deployed using any existing software deployment systems including Active Directory and SMS. ThinApp has no client or server components to manage or maintain and ThinApp can transparently stream large applications from any network attached storage devices without server software.
- **Sandboxing prevents modifications**—ThinApp redirects all changes intended for the host computer's file system and registry to a private per-user sandbox. Sandboxes can be located on a network share, allowing application settings to follow users as they move from machine to machine. For mobile users, sandboxes can be stored on local USB flash drives, thus preventing damage to the host computer or accidental host storage of sensitive data.

About the ThinApp Assistant

Information about the ThinApp Assistant is organized into the following sections:

- [Process for Authoring a ThinApp Application Using the ThinApp Assistant](#)
- [Components of a ThinApp Application](#)
- [Supported InstallShield Project Types](#)
- [How Transforms are Included in a ThinApp Application](#)
- [About Sandboxes](#)

Process for Authoring a ThinApp Application Using the ThinApp Assistant

You can use the ThinApp Assistant to convert a Windows Installer package into a ThinApp application. During this process, you:

- **General Settings**—Specify sandbox and Active Directory settings.
- **Files, Folders, Shortcuts, Registry Settings**—Specify the files, folders, shortcuts, and registry settings that will be included in the ThinApp application.
- **Isolation Options**—Override the default isolation options for selected folders and registry keys.
- **Build**—Specify build options and build a ThinApp application.

The following diagram illustrates the ThinApp application creation process:

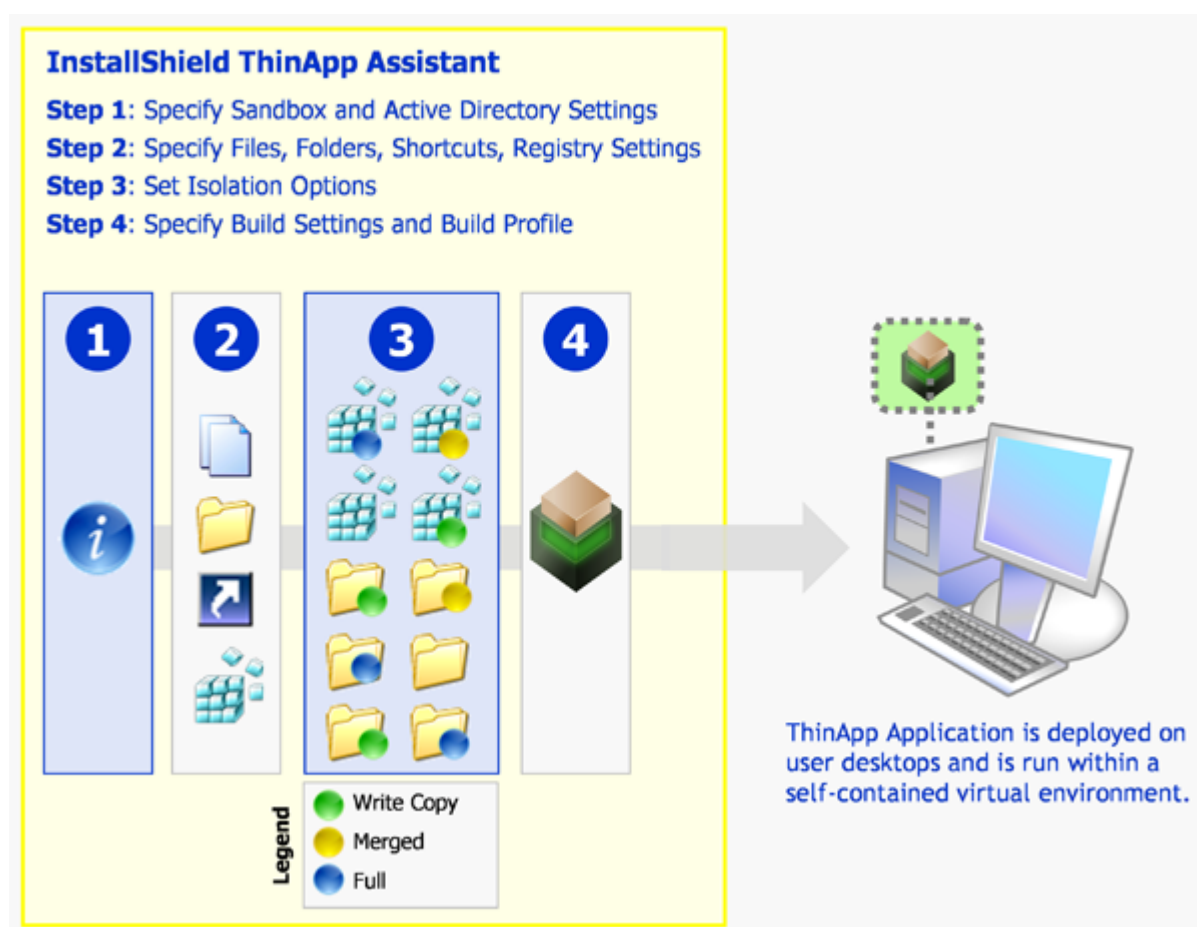


Figure 10-27: Creating a ThinApp Application



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

The process for authoring a ThinApp application using the ThinApp Assistant is as follows:

Table 10-23 • Steps to Convert a Windows Installer Package to a ThinApp Application







Step	Go To:	Actions
Getting Started	InstallShield Start Page	Create or open one of the following project types: <ul style="list-style-type: none"> • Basic MSI • MSI Database (Direct Edit Mode) • Transform (Direct MST Mode)
	InstallShield Start Page	Click on the VMware ThinApp tab to open the ThinApp Assistant Home page.
	ThinApp Assistant Home Page	Click General Settings in the navigation bar to open the General Settings page.
Specifying ThinApp General Settings	General Settings Page 	Specify the sandbox name and sandbox options for the ThinApp application, control access to the ThinApp application via Active Directory, and specify whether to include diagnostic tools with the ThinApp application.
Managing Files and Folders in a ThinApp Application	Files & Folders Page 	View existing files and folders, add and delete files.
Setting ThinApp Isolation Options	Files & Folders Page 	Override the default isolation options for selected folders. Isolation options specify how the virtual environment will provide access to folders requested by the ThinApp application.
Modifying Shortcuts to the ThinApp Application's Executable Files	Applications Page 	Create, delete, include, exclude, or rename ThinApp application executables, which are derived from the shortcuts in its Windows Installer package.
Modifying ThinApp Application Registry Settings	Registry Page 	Add, delete, or modify the registry settings in your ThinApp application, and override the default isolation options for selected registry keys. Isolation options specify how the virtual environment will provide access to registry keys requested by the ThinApp application.

Table 10-23 • Steps to Convert a Windows Installer Package to a ThinApp Application

Step	Go To:	Actions
Modifying Build Options	Build Options Page 	<p>[Basic MSI Project mode] Select the releases that you want to build.</p> <p>[Direct Edit or Direct MST mode] To enable the Build function for a ThinApp application, select the Build ThinApp application option.</p>
Building a ThinApp Application	Build on the Toolbar OR Build Virtual Package Button	<p>Click Build to build the active Release and create a ThinApp application.</p> <p>When you are in Direct Edit mode, click the Build Virtual Package button to save the Windows Installer package and create a ThinApp application.</p>



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

Components of a ThinApp Application

When you use the ThinApp Assistant to build a ThinApp virtual package, the resources you generate are called ThinApp applications. The number of files included in a ThinApp application depends upon how many shortcuts are defined in the project (or Windows Installer package) and whether you choose to include diagnostic tools with the ThinApp application.

Table 10-24 • Components of a ThinApp Application









Number of Shortcuts	ThinApp Application Components	Description
1 shortcut	ProductName.exe  AdminMaster70.EXE AdminMaster IT Toolz, Inc.	<p>The ThinApp application consists of a single executable (.exe) file:</p> <ul style="list-style-type: none"> • Launching the application—This executable file is used to launch the ThinApp application. • Location of application data—This executable file contains all of the files, registry keys, DLLs, ThinApp components, and third party libraries that are required for the application to run.

Table 10-24 • Components of a ThinApp Application

Number of Shortcuts	ThinApp Application Components	Description
More than 1 shortcut	ProductName.exe FeatureName.exe Package.DAT  XYZPhotoBrowse40.EXE XYZ Photo Browse XYZ Software, Inc.  XYZPhotoTouchUp40.EXE XYZ Photo TouchUp XYZ Software, Inc.  Package.DAT DAT File 128,253 KB	The ThinApp application consists of two or more executable files and a Package.DAT file: <ul style="list-style-type: none"> • Launching the application—Each of the executables is used to launch the ThinApp application or a specific feature of the ThinApp application. • Location of application data—The Package.DAT file contains all of the files, registry keys, DLLs, ThinApp components, and third party libraries that are required for the application to run.
1 shortcut with diagnostic tools	ProductName.exe cmd.exe regedit.exe Package.DAT  AdminMaster70.EXE AdminMaster IT Toolz, Inc.  cmd.exe Windows Command Processor Microsoft Corporation  regedit.exe Registry Editor Microsoft Corporation  Package.DAT DAT File 128,253 KB	The ThinApp application consists of three executable files and a Package.DAT file: <ul style="list-style-type: none"> • Launching the application—The package executable is used to launch the ThinApp application. • Launching the diagnostic tools—The cmd.exe and regedit.exe executables are used to launch the Command Prompt and Registry Editor diagnostic tools. • Location of application data—The Package.DAT file contains all of the files, registry keys, DLLs, ThinApp components, and third party libraries that are required for the ThinApp application and the diagnostic tools to run.

ThinApp application files are saved in a directory named ThinAppPackage. The location of the ThinAppPackage directory depends upon the type of file you are editing in InstallShield:

- **InstallShield project**—The ThinAppPackage directory will be located in a subdirectory of the directory that contains this InstallShield project file, such as:
 C:\InstallShield 2009 Projects\ProductName\ConfigurationName\ReleaseName\ThinAppPackage
- **Windows Installer package**—The ThinAppPackage directory will be located in the same directory as the Windows Installer file, such as:
 C:\FolderContainingMSI\ThinAppPackage



Caution • Modifying these files directly is **not recommended**. To make any modifications, use the InstallShield ThinApp Assistant.

Intermediate Data Files: Interm Directory

When a ThinApp application is built, files that support the ThinApp application build process are extracted out of the Windows Installer package and saved in a subdirectory of the ThinAppPackage directory named the Interm directory.



Figure 10-28: Interm Subdirectory of the ThinAppPackage Directory

The data in this directory is then compiled into ThinApp application as part of the build process. The data in the Interm directory *does not* need to be distributed with the ThinApp application.

Supported InstallShield Project Types

The **VMware ThinApp** tab is available when one of the following InstallShield project types is open:

- Basic MSI Project
- MSI Database (Direct Edit Mode)
- Transform (Direct MST Mode)

How Transforms are Included in a ThinApp Application

The ThinApp Assistant supports the inclusion of transform files with Windows Installer packages in a ThinApp application.

- **How transforms are applied when building a ThinApp application**—When building a ThinApp application, transforms that you have specified are automatically applied to the base Windows Installer (.msi) package to create a temporary package, and then the ThinApp application is generated from that temporary package.
- **Creating a new transform**—You can create a new transform in InstallShield, and then build a ThinApp application from that transform file. When you create a new transform file in InstallShield, you specify the root .msi file in the **Open Transform** wizard. The steps you take to generate a ThinApp application after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.
- **Converting a Windows Installer package with existing transforms**—If you have a Windows Installer package and one or more existing transform files, and you want to include these transforms in the ThinApp application, you need to open one of the *transforms* in InstallShield (rather than the .msi file). The **Open Transform** wizard will open, and you will be prompted to specify the root .msi file and which of the existing .mst files you want to include. The steps you take to generate a ThinApp application after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.



Caution • All of the transforms that you add to a ThinApp application must be located in the same folder as the Windows Installer .msi package so that they can be accessed when the ThinApp application is built.

About Sandboxes

A ThinApp application runs in a *sandbox*, a virtual operating system—consisting of a virtual file system and a virtual registry—which is embedded with each ThinApp application. Running an application in a sandbox protects the local operating system from installation modifications that could affect stability or security. In a sandbox, system resources (such as files and registry keys) are redirected from the physical operating system files to the sandbox.

Many applications fail to run if the user does not have administrative rights because they expect to be able to write to global locations like HKEY_LOCAL_MACHINE and C:\Program Files. Using sandbox technology makes applications believe they have the ability to make global changes when they are actually writing to user and application-specific locations, and allows applications that require administrative rights to run without additional privileges. This feature allows ThinApp applications to run in security-restricted environments such as Windows Vista and Terminal Server.

What is a Sandbox Cache?

When you run a ThinApp application, additional files or registry keys may be produced. Depending upon the isolation options, some of this run time data will need to be stored locally in a sandbox cache, a local per-user directory.

When the ThinApp application is built, the *local Sandbox cache* is created in the following location, using the **Sandbox Name** that was entered on the **General Settings** page.

c:\Documents & Settings\USER_NAME\Application Data\ThinApp\SANDBOX_NAME

If the user's Application Data directory is stored on a network share, the ThinApp application's settings will automatically migrate when the same user logs in on another machine. You can also choose to create the sandbox cache in an external storage device such as a USB flash drive.

Using the ThinApp Assistant to Create a ThinApp Application

The steps you need to take to create a ThinApp application are the following:

Table 10-25 • Steps to Take to Create a ThinApp Application Using the ThinApp Assistant

Step #	Description
Step 1	Specifying ThinApp General Settings
Step 2	Managing Files and Folders in a ThinApp Application
Step 3	Setting ThinApp Isolation Options
Step 4	Modifying Shortcuts to the ThinApp Application's Executable Files
Step 5	Modifying ThinApp Application Registry Settings
Step 6	Modifying Build Options

Table 10-25 • Steps to Take to Create a ThinApp Application Using the ThinApp Assistant

Step #	Description
Step 7	Building a ThinApp Application

Specifying ThinApp General Settings

When creating a ThinApp application, you can, optionally, specify sandbox and Active Directory settings. You can also specify whether to include diagnostic tools with the ThinApp application. The following tasks are performed on the **General Settings** page of the **ThinApp Assistant**:

- [Specifying Sandbox Information](#)
- [Specifying Control Access via Active Directory](#)
- [Prerequisites for Building a ThinApp Application](#)
- [Including Diagnostic Tools With an App-V Application](#)

Specifying Sandbox Information

In this step, you have the option of entering a name for the [Sandbox cache](#) that is created when the ThinApp application is built.



Note • For information on sandboxes and sandbox caches, see [About Sandboxes](#).



Task: [To specify sandbox information:](#)

1. In the **ThinApp Assistant**, open the **General Settings** page.
2. When a ThinApp application is built, a [Sandbox cache](#) is created in the following location:

c:\Documents & Settings\USER_NAME\Application Data\ThinApp\SANDBOX_NAME

By default, AdminStudio names the Sandbox by assigning it a unique GUID. However, if you want to override this default Sandbox name, you may (optionally) enter a new name in the **Sandbox Name** field.



Note • The **Sandbox Name** you enter here is also recorded in the *Package.ini* file.

3. If you want changes for Network mapped drives to be saved in the sandbox, then select the **Mapped Network Drive Changes go to Sandbox** option.

4. If you want changes for removable disks to be saved in the sandbox, then select the **Removable Disk Changes go to Sandbox** option.
5. If you want to delete the sandbox content when the ThinApp application exits, then select the **Reset Sandbox on Exit** option.

Specifying Control Access via Active Directory

You can control the access of users to a ThinApp application by specifying Active Directory groups on the **General Settings** page. At build-time, ThinApp assigns a unique GUID-like number to uniquely identify each Active Directory Group that you identify. Members of those groups will have access to the ThinApp application. For more detailed information about how Active Directory permissions are assigned, see [About Controlling Access to ThinApp Applications](#).

To specify control access via Active Directory on the **General Settings** page, perform the following steps:



Task: To specify control access via Active Directory:

1. In the **ThinApp Assistant**, open the **General Settings** page.
2. Select the **Control Access via Active Directory** option. The fields below are enabled.

3. In the **Allow application execution to the following user groups** field, enter the names of all of the Active Directory groups that you want to have permission to run this ThinApp application, separated by semi-colons, such as:

GroupOne;GroupTwo;GroupThree
4. In the **Message shown when users not belonging to above groups run the ThinApp application**, enter the message that will be displayed when users that do not belong to the specified groups attempt to launch a ThinApp application.



Caution • If you do not select the **Control Access via Active Directory** option, anyone who has access to a directory containing a ThinApp application will be able to run the application.

About Controlling Access to ThinApp Applications

Note the following about controlling access to ThinApp applications via Active Directory:

- **You must be connected**—You must be connected to your Active Directory domain when you build the ThinApp application.
- **Groups must exist**—The Active Directory groups that you specify must exist when the ThinApp application is built.
- **If you delete a group and then recreate it, you must rebuild**—If you delete a group and recreate it, you will need to rebuild the ThinApp application in order to authenticate against the “new” group.
- **Offline users can authenticate using cached credentials**—When users are offline, they can authenticate using cached credentials. Assuming that the user can log into their laptop, ThinApp Active Directory authentication will still work.
- **Sometimes you may need to update credentials manually**—Cached credentials may not refresh on clients until the next Active Directory refresh cycle. To manually refresh the cached group policy credentials, you can use the gpupdate command. Sometimes the user may need to log-off before the credentials are recached.
- **“Administrators” and “Everyone” Groups use same credentials**—Special groups like Administrators and Everyone have the same SID on every Active Directory domain and Workgroup. Other groups you create will have a domain-specific SID, meaning a user cannot create their own local group with the same name to bypass authentication.

Prerequisites for Building a ThinApp Application

AdminStudio will convert the package installation into a format compatible with ThinApp. However, the ThinApp build process requires the availability of certain ThinApp tools.

As a prerequisite to building a ThinApp application from AdminStudio, you must have installed ThinApp and accepted any and all license agreements.



Note • For more information, see [ThinApp](#) on the VMware Web site.

Including Diagnostic Tools With a ThinApp Application

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **General Settings** page, you can choose to include the Registry Editor and the Windows Command Prompt diagnostic tools with your ThinApp application.

If you include diagnostic tools with your ThinApp application, you will be able to look at the registry or file system for the application while it is running in its virtual environment. For example, if you were running a ThinApp application and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.



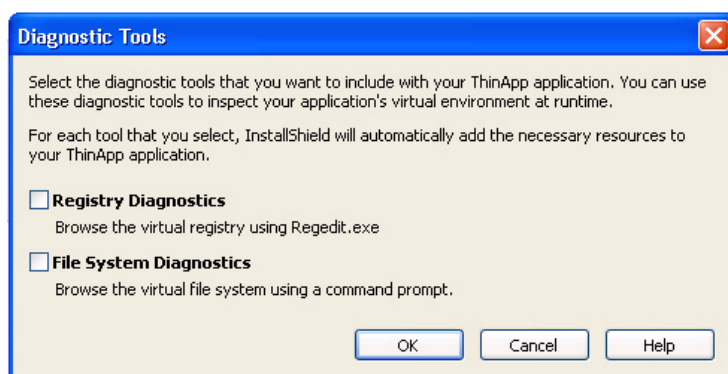
Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the ThinApp application. However, these tools may not be compatible with other operating systems.



Task:

To include diagnostic tools with a ThinApp application:

1. In the **ThinApp Assistant**, open the **General Settings** page.
2. In the **More Options** list, click **Diagnostic Tools**. The **Diagnostic Tools** dialog box opens.



3. If you want to include the Registry Editor (`regedit.exe`) with your ThinApp application so that you can browse the registry at runtime from within the virtual environment, select the **Registry Diagnostics** option.
4. If you want to include the Windows Command Prompt application with your ThinApp application so that you can browse the virtual file system at runtime from within the virtual environment, select the **File System Diagnostics** option.

Launching the Diagnostic Tools Within the Virtual Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the ThinApp application.

When the user runs this ThinApp application, two additional shortcuts will be available in the application's shortcut folder: The names of these shortcuts will reflect the application name, such as:

```
[ProductName] Registry  
[ProductName] File System
```

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application's virtual environment.

Managing Files and Folders in a ThinApp Application

The next step in creating a ThinApp application is to view existing files and folders, add and delete files and folders, and override the default isolation options for folders.

The following tasks are performed on the **Files & Folders** page.

- [Adding, Deleting, and Moving Files and Folders in a ThinApp Application](#)
- [Controlling the Display of Predefined Folders](#)

Adding, Deleting, and Moving Files and Folders in a ThinApp Application

The directories in the destination tree on the **Files & Folders** page of the ThinApp Assistant represent how your application will look when it is installed on to your customer's machine.

On the **Files & Folders** page, you can view all of the files and folders that are currently in your ThinApp application, add new files and folders to include in the ThinApp application, and delete files and folders from the ThinApp application.

- [Adding Files to a ThinApp Application](#)
- [Adding an Existing Folder \(and its Contents\) to a ThinApp Application](#)
- [Creating a New Folder](#)
- [Moving Files and Folders](#)
- [Deleting Files and Folders](#)

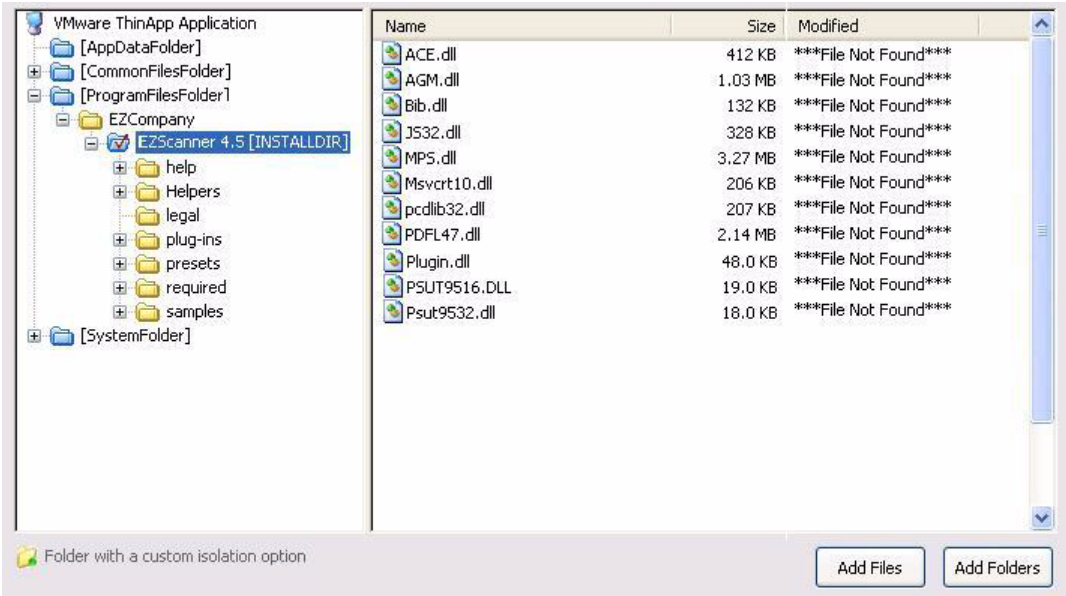
Adding Files to a ThinApp Application

To add files to a ThinApp application, perform the following steps:



Task: *To add a files to a ThinApp application:*

1. In the **ThinApp Assistant**, open the **Files & Folders** page. The files and folders are listed in the **VMware ThinApp Application** tree, organized by installation directory.



- Folders are listed in the column on the left, and all of the files in the selected folder are listed on the right. Blue folders are the supported MSI standard folders. The folder with the check mark is **INSTALLDIR**, which represents the main product installation directory.
2. Browse through the folder tree to find the folder that you would like to add files to.
 3. Select the folder and click the **Add Files** button. The **Open** dialog box opens.
 4. Select the file or files that you want to add and click **Open**. The files you selected are now listed.



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Adding a File by Dragging and Dropping Files From Your System

You can also add files or folders to your ThinApp application on the **Files & Folders** page by dragging them from a directory on your computer to the desired location in the tree.

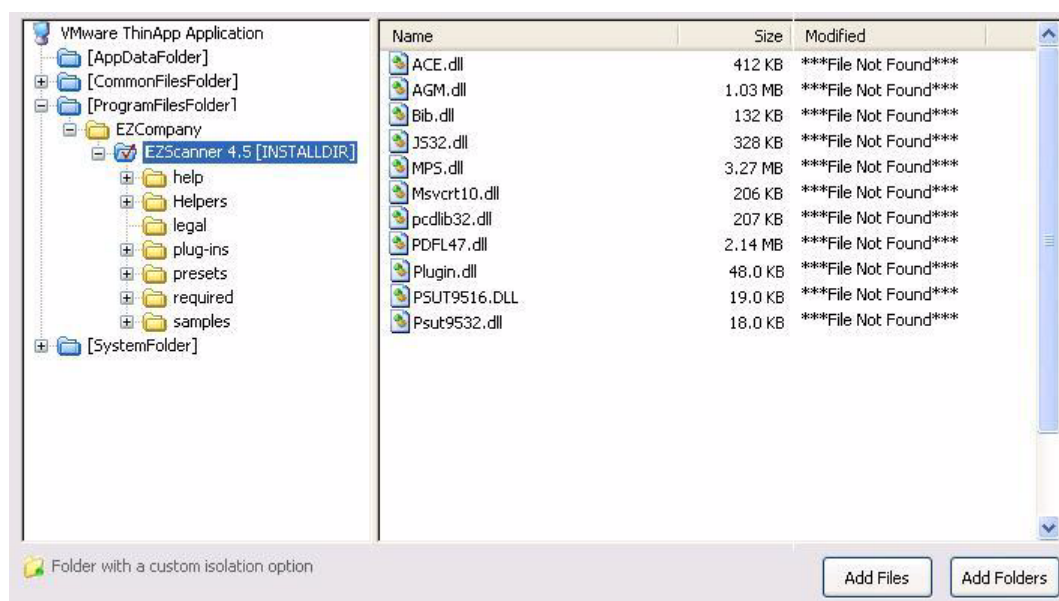
Adding an Existing Folder (and its Contents) to a ThinApp Application

To add an existing folder and all of the files and subfolders within it to a ThinApp application, perform the following steps:

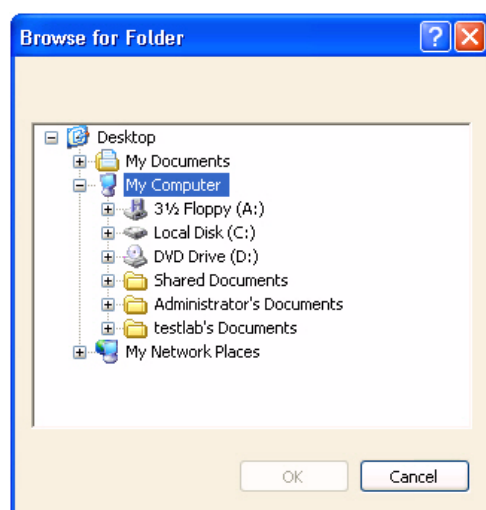


Task: *To add an existing folder to a ThinApp application:*

1. In the **ThinApp Assistant**, open the **Files & Folders** page. The files and folders are listed in the **ThinApp Application** tree, organized by installation directory.

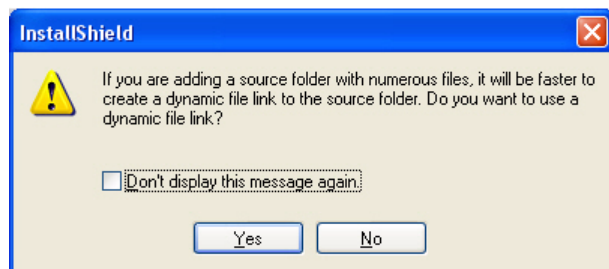


2. Browse through the folder tree to find the folder that you would like to add a folder into.
3. Select the folder and click the **Add Folders** button. The **Browse for Folder** dialog box opens, listing all of the directories available to your computer.



4. Select a folder and click **OK**.

If you are editing an InstallShield project (not a Windows Installer package), you are prompted to choose whether you want to create a dynamic file link to the source folder.



5. Indicate whether you want to create a dynamic file link by selecting one of the following:
 - **No**—For more flexibility with ThinApp options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
 - **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See Dynamic File Link Settings Dialog Box.

The folder that you selected is now listed, along with of the files and folders within it.

Creating a New Folder

You can create a new, empty folder by selecting an existing folder in the tree and selecting **New Folder** from the context menu.



Task: To create a new folder:

1. Right-click on a folder in the **VMware ThinApp Application** tree and select **New Folder**. A new folder is created as a subfolder of the selected folder:



2. Enter a name for the new folder.

Moving Files and Folders

To change the folder's location in the ThinApp application folder tree structure, perform the following steps:



Task: *To move a file or folder:*

1. Select the file or folder that you want to move.
2. With the mouse button down, drag the file or folder to the new location.
3. Release the mouse button.

Deleting Files and Folders

To delete a file or a folder (and all of its contents) from a ThinApp application, perform the following steps:



Task: *To delete a file or folder:*

1. Select the file or folder in the **VMware ThinApp Application** tree that you want to delete.
2. Select **Delete** from the context menu. You are prompted to confirm the deletion.
3. Click **Yes**. The selected file or folder is deleted.



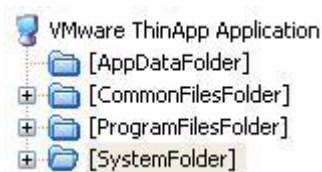
Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains from the entire Project, not just from the ThinApp application.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).

Controlling the Display of Predefined Folders

On the **Files & Folders** page, the **VMware ThinApp Application** tree initially displays the more commonly used predefined folders, such as [ProgramFilesFolder] and [CommonFilesFolder].



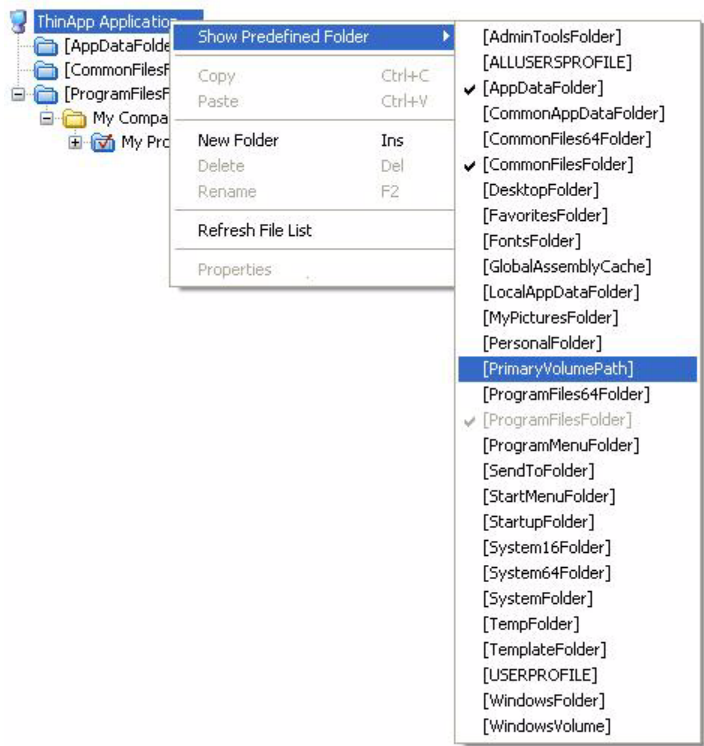
These predefined folders are dynamic, meaning that they do not use hard-coded paths. The value for each destination folder is obtained from the operating system of the target machine.

You can control which predefined folders are listed in this tree.



Task: *To change which predefined folders are listed:*

1. In the **VMware ThinApp Application** tree, select the **ThinApp Application** node (or any of the files or folders that are listed, point to **Show Predefined Folder**. A list of predefined folders opens.



Those folders that are already displayed are preceded by a check mark, and those that are not displayed do not have a check mark.

2. To add a folder to the tree listing, select a folder that is not currently listed in the tree.



Note • These predefined folders are always added to the root of the **VMware ThinApp Application** tree, no matter what file or folder you had selected when you selected it from the Predefined Folders list.

3. To remove a folder from the tree listing, select that folder name in this list (which is preceded by a check mark).



Note • You cannot turn off the display of the [ProgramFilesFolder].

Setting ThinApp Isolation Options

ThinApp uses a sandbox virtual environment to control application compatibility and accessibility. The isolation option that is assigned to a folder or registry key specifies how the virtual environment will provide access to system resources requested by the application. You can use isolation options to control what an application can read and write on the local machine.

The default settings for isolation options are built into the ThinApp Assistant, and those defaults are adequate for most environments. However, in the ThinApp Assistant, you can override the default settings for selected folders or registry keys to exert control over application interactions with client operating system resources.

You set isolation options on the **Isolation Options** dialog box, which is open by selecting a folder or registry key and then selecting **Isolation Options** from the context menu.

Information about setting isolation options is presented in the following topics:

- [Overview of ThinApp Isolation Options](#)
- [Setting Isolation Options for Folders](#)
- [Inheritance of Isolation Options from Folders to Files](#)



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, ThinApp, and registry settings.*

Overview of ThinApp Isolation Options

ThinApp uses virtual environments to control application compatibility and accessibility. The *isolation option* that is assigned to a folder or registry key specifies how the virtual environment will provide access to system resources requested by the application.

The default settings for isolation options are built into the ThinApp Assistant, and those defaults are adequate for most environments. However, in the ThinApp Assistant, you can override the default settings for selected folders or registry keys to exert control over application interactions with client operating system resources.



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, ThinApp, and registry settings. Select the **Default** isolation option unless you require specific custom handling.*

You set isolation options on the **Isolation Options** dialog box, which is opened by selecting **Isolation Options** on the context menu when you have a folder selected on the **Files & Folders** page or a registry key selected on the **Registry** page.

Information about isolation options is presented in the following sections:

- [Available ThinApp Isolation Options](#)
- [ThinApp Isolation Option Use Scenarios](#)
- [ThinApp Assistant Default Isolation Options](#)

Available ThinApp Isolation Options

On the **Isolation Options** dialog box, you can choose one of the following isolation options:

Table 10-26 • ThinApp Isolation Options

Option	Visibility of System Elements	Modifications to Virtual Elements	Modifications to System Elements	New Elements	If System and Virtual Element at Same Location
Default	<i>As defined internally by the ThinApp Assistant</i>				
Write Copy	Visible	Sandbox	Sandbox	Created in Sandbox	Sees Virtual Element
Merged	Visible	Sandbox	System	Created in System	Sees Virtual Element
Full	Not Visible	Sandbox	N/A (System elements cannot be modified)	Created in Sandbox	N/A (System elements cannot be read)

ThinApp Isolation Option Use Scenarios

The following table describes scenarios where you would use each isolation option:

Table 10-27 • Use Scenarios for ThinApp Isolation Options

Option	Use Scenario
Write Copy	<p>You would use Write Copy isolation when:</p> <ul style="list-style-type: none">• Application was not designed or tested for multi-user environments and expects it can modify files and keys without impacting other users.• Application expects write permission to Global locations and was not designed for locked-down desktop environments found in corporate environments or Windows Vista. <p>With Write Copy isolation, ThinApp makes copies of registry keys and files written by the application and performs all of the modifications in a user-specific sandbox. With this type of isolation, the ThinApp applications believe that they have global write permissions, while they really only modify the sandbox directory.</p>
Merged	<p>You would use Merged isolation when the ThinApp application needs write access to user-specific storage areas, like the Desktop and My Documents.</p>

Table 10-27 • Use Scenarios for ThinApp Isolation Options

Option	Use Scenario
Full	<p>You would use Full isolation when a ThinApp application needs to run on a machine where earlier or later versions of the same application are either installed or were not uninstalled correctly.</p> <p>For directories and registry keys that have Full isolation, the ThinApp application will not be aware of any host computer file that might exist, and it sees only virtual files and registry keys at fully isolated locations.</p>

ThinApp Assistant Default Isolation Options

If you do not set any isolation options on a folder or registry key in the ThinApp Assistant, the following default isolation options are applied:

Table 10-28 • ThinApp Assistant Default Isolation Options

Isolation Option	Condition
Write Copy Isolation	All other directories and subkeys associated with the product are assigned Write Copy isolation.
Merged Isolation	User-specific storage areas like the Desktop and My Documents, are set to Merged Isolation so that application has direct Write access to these locations



Note • Network shares are not affected by isolation modes. Read and write operations to network shares occur unchanged by ThinApp.



Note • These default isolation options are built into the ThinApp Assistant.

Setting Isolation Options for Folders

To override a folder's default isolation options, perform the following steps:



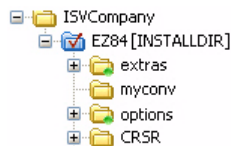
Task: *To set an isolation option on a folder.*

1. Open the **Files & Folders** page.
2. Browse through the folder tree to find the folder that you would like to modify.
3. Select the folder and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.

Chapter 10: Creating Customized Virtual Applications

Creating ThinApp Applications

4. Select one of the following options, as described in [Available ThinApp Isolation Options](#).
 - Default
 - Write Copy
 - Merged
 - Full
5. Click **OK**. Folders that have an isolation setting other than default are marked with a special icon:



Inheritance of Isolation Options from Folders to Files

Isolation options for files and subfolders are always inherited. The ThinApp virtual environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for C:\Windows and one for C:\Windows\System32. When the application requests C:\Windows\System32\notepad.exe, then the C:\Windows\System32 isolation rule will be applied because C:\Windows\System32 is a more specific reference to C:\Windows\System32\notepad.exe than is C:\Windows.

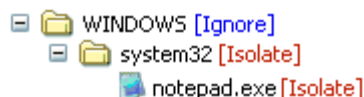


Figure 10-29: Example of Inheritance of Isolation Options from Folders to Files

Modifying Shortcuts to the ThinApp Application's Executable Files

You define application shortcuts to enable users to launch a ThinApp application from within the virtual environment.

By default, the **ThinApp Assistant** creates ThinApp applications for all of the executable shortcuts that exist in your project (or Windows Installer package). These shortcuts are listed in a checklist on the **Applications** page.

When you select each shortcut, details about it are displayed:

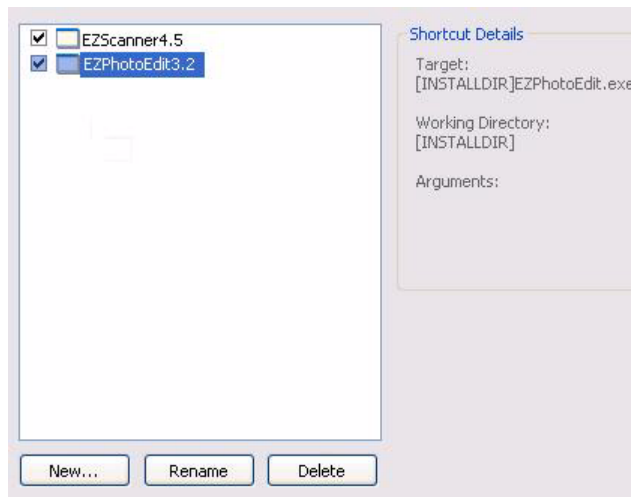


Figure 10-30: List of Shortcuts for an Application



Caution • You must define at least one shortcut to enable users to launch the application from the isolation environment.

On the **Applications** page, you can create, delete, include, exclude, or rename ThinApp application executables, which are derived from the shortcuts in its Windows Installer package.

- [ThinApp Applications and the Virtual Environment](#)
- [Compressing a ThinApp Application](#)
- [ThinApp Shortcut Requirements](#)
- [Creating a New ThinApp Application](#)
- [Including an Existing ThinApp Application](#)
- [Excluding or Deleting an Existing ThinApp Application](#)
- [Excluding vs. Deleting ThinApp Application Shortcuts](#)
- [Renaming a ThinApp Application](#)



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

ThinApp Applications and the Virtual Environment

On the **Applications** page of the ThinApp Assistant, you define application shortcuts to enable users to launch a ThinApp application from within the virtual environment. By default, the ThinApp Assistant creates ThinApp applications for all of the executable shortcuts that exist in your project.

To deploy a ThinApp application—on a local drive or a network share—systems administrators just need to give users access to the ThinApp application.

Compressing a ThinApp Application

- A ThinApp application consists of either:
- **One executable file (.exe)**—This file is used to both launch the ThinApp application and also contain all of the data that is required for the application to run. In this scenario, this executable file would be a large file.
- or
- **Several executable files (.exe) and a Package.DAT file**—Each of the executables is used to launch the ThinApp application or a specific feature of the ThinApp application and Package.DAT contains all of the data that is required for the application to run. In this scenario, Package.DAT would be a large file.

Each time a user launches a ThinApp application, its data (from either the executable file or from Package.DAT) is read into the computer’s memory. To reduce the application size, you can select a **Compression Type** on the **Build Options** page to compress all of the data.

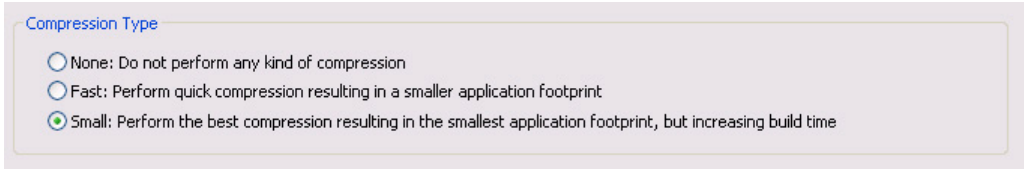


Figure 10-31: Compression Types on the Build Options Page

The following **Compression Types** are available:

Table 10-29 • ThinApp Compression Types

Type	Description
None	Do not perform any type of compression.
Fast	Perform quick compression resulting in a smaller application footprint.
Small	Perform the best compression resulting in the smallest application footprint, but increasing build time.

Application startup time is most effected by compression options used:

- **No compression**—Without compression enabled, startup speeds are comparable to normal application startup times.
- **Fast compression**—With fast compression options enabled, applications may startup faster than normal when the disk cache is empty and slightly slower than normal when the disk cache has been pre-filled, depending on processor speed and disk speeds.

You may also want to compress a ThinApp application to make it easier to distribute it throughout your organization.

When you perform compressed builds, large temporary files are saved in a cache location. To delete all of these temporary files, select the **Clear the VMware ThinApp Cache** option in the **More Options** list on the **Build Options** page

ThinApp Shortcut Requirements

For each ThinApp application, you are required to define **at least one** shortcut. You define application shortcuts to enable users to launch a ThinApp application from within the virtual environment. By default, the ThinApp Assistant creates ThinApp applications for all of the executable shortcuts that exist in your project (or Windows Installer package).

If you build a ThinApp application that does not contain any shortcuts, users will not be able to launch the application.

Creating a New ThinApp Application

On the **Applications** page of the ThinApp Assistant, you specify the executables that you want to create ThinApp applications for.



Task: *To create a new ThinApp application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project (or Windows Installer package) are listed:
 - Those that are currently included in the ThinApp application are selected.
 - Those that are currently excluded from the ThinApp application are not selected.
2. Click **New**. The **Browse for a Shortcut Target File** dialog box opens and prompts you to select a file within this ThinApp application.
3. Select the file that you want to create a shortcut to.
4. Click **Open**. A new shortcut is listed, and it is named the same name as the selected file.
5. To include this shortcut in the ThinApp application, make sure that its check box is selected.

Including an Existing ThinApp Application

If you want to include a previously excluded shortcut in a ThinApp application, perform the following steps:



Task: **To include an existing ThinApp application:**

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To include a previously excluded shortcut, select the shortcut and select the check box.

Excluding or Deleting an Existing ThinApp Application

By default, the ThinApp Assistant creates ThinApp applications for all of the executable shortcuts that exist in your project (or Windows Installer package). These shortcuts are listed in a checklist on the **Applications** page.

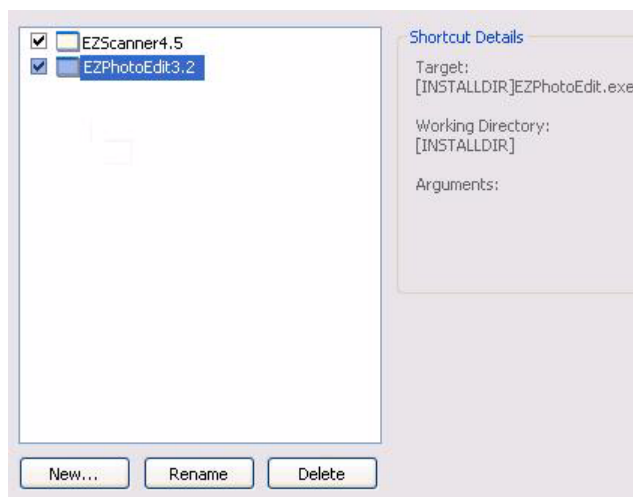


Figure 10-32: Initial List of Shortcuts for an Application

To prevent the shortcut from being created in the ThinApp application, you can choose to either delete or exclude it.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the ThinApp application, but it will remain in the InstallShield project. See [Excluding a ThinApp Application](#).
- **Deleting a shortcut**—When you delete a shortcut, it is removed from both the ThinApp application and the InstallShield project. See [Deleting a ThinApp Application](#).



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

If you have any unnecessary shortcuts in your project, you can simply exclude them from the ThinApp application by unchecking them in the shortcuts list. If you like to permanently remove a shortcut, you can delete it from the shortcut list.

Excluding a ThinApp Application

If you want to exclude one of these shortcuts from being created in the ThinApp application, perform the following steps:



Task: *To exclude a ThinApp application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To exclude a shortcut, select the shortcut and clear the check box.



Note • When you exclude a shortcut, it will not be created in the ThinApp application, but it will remain in the InstallShield project.

Deleting a ThinApp Application

To delete a ThinApp application, perform the following steps.



Task: *To delete a ThinApp application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
2. Select the shortcut and click **Delete**.



Caution • If you delete a shortcut on the **Applications** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

Excluding vs. Deleting ThinApp Application Shortcuts

To prevent a shortcut from being created in the ThinApp application, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the ThinApp application, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding a ThinApp Application](#).
- **Deleting a shortcut**—When you delete a shortcut, it is removed from both the ThinApp application and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting a ThinApp Application](#).

Renaming a ThinApp Application

To rename a ThinApp application, perform the following steps:



Task: *To rename a ThinApp application:*

1. Open the **Applications** page. All of the shortcuts that exist in the project are listed.
2. Select the shortcut that you want to rename and click **Rename**. A box appears around the shortcut name, and the shortcut name becomes an editable field.
3. Enter a new name for the shortcut.

Modifying ThinApp Application Registry Settings

Using the **ThinApp Assistant**, you can view existing registry keys, values, and data, and add or delete registry items in your ThinApp application.

You can also override the default isolation options for selected registry keys. Isolation options specify how the virtual environment will provide access to system resources requested by the application.

Information about modifying registry settings on the **Registry** page includes the following topics:

- [About the Windows Registry](#)
- [Adding or Deleting Registry Keys and Values](#)
- [Setting ThinApp Isolation Options on Registry Keys](#)

About the Windows Registry

The Windows registry is a system-wide database that contains configuration information used by applications and the operating system. The registry stores all kinds of information, including the following:

- Application information such as company name, product name, and version number
- Path information that enables your application to run

- Uninstallation information that enables end users to uninstall the application easily without interfering with other applications on the system
- System-wide file associations for documents created by an application
- License information
- Default settings for application options such as window positions

Keys, Value Names, and Values

The registry consists of a set of keys arranged hierarchically under the My Computer explorer. Just under My Computer are several root keys. An installation can add keys and values to any root key of the registry. The root keys that are typically affected by installations are:

- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT

A key is a named location in the registry. A key can contain a subkey, a value name and value pair, and a default (unnamed) value. A value name and value pair is a two-part data structure under a key. The value name identifies a value for storage under a key, and the value is the actual data associated with a value name. When a value name is unspecified for a value, that value is the default value for that key. Each key can have only one default (unnamed) value.

Note that the terms key and subkey are relative. In the registry, a key that is below another key can be referred to as a subkey or as a key, depending on how you want to refer to it relative to another key in the registry hierarchy.

Adding or Deleting Registry Keys and Values

Editing the registry on the **Registry** page is performed much like it is performed on the InstallShield **Registry View**. See [Editing the Registry](#).

Setting ThinApp Isolation Options on Registry Keys

To override a registry key's default isolation options (which are built into the ThinApp Assistant), perform the following steps:



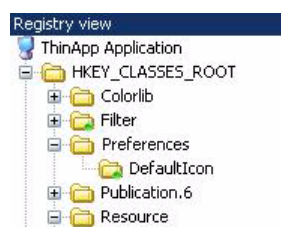
Task: *To set an isolation option on a registry key:*

1. Open the **Registry** page.
2. Browse through the registry tree to find the key that you would like to modify.
3. Select the folder or key and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.



Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

4. Select one of the following options, as described in [Available ThinApp Isolation Options](#).
 - Default
 - Write Copy
 - Merged
 - Full
5. Click **OK**. Registry keys that have an isolation setting other than default are marked with a special icon:



Tip • To import an existing registry (.reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.

Inheritance of ThinApp Isolation Options in the Registry

Isolation options for registry keys are always inherited. The ThinApp virtual environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for the Microsoft registry key and one for Microsoft\Windows registry key. When the application requests Microsoft\Windows\CurrentVersion, then the Microsoft\Windows isolation rule will be applied because Microsoft\Windows is a more specific reference to Microsoft\Windows\CurrentVersion than is Microsoft.

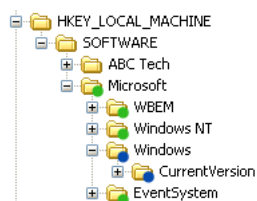


Figure 10-33: Example of Inheritance of Isolation Options from Folders to Files

Modifying Build Options

On the **Build Options** page, you choose which releases of this InstallShield project you want to build a ThinApp application for when the project is built, specify the type of compression, disable the ThinApp Log Monitor tracing capabilities, and specify whether you want to include additional Windows Installer packages in the virtual package.

Also, if you are editing a Windows Installer package in Direct Edit mode (or Direct MST mode), you need to select the **Build ThinApp Application** option on the **Build Options** page before you will be able to build a ThinApp application for that Windows Installer package.

- [Selecting Releases to Build](#)
- [Enabling ThinApp Application Building When in Direct Edit Mode](#)
- [Including Additional Windows Installer Packages in a ThinApp Application](#)
- [Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application](#)
- [Setting ThinApp Log Monitor Tracing Options](#)
- [Setting AppLink Options](#)
- [Setting AppSync Options](#)



Important • You must create at least one Release (on the **Releases** view of the Installation Designer) before you will be able to select a Release on the **Build Options** page.

Selecting Releases to Build

You select the releases that you want to build a ThinApp application for on the **Releases** tree of the **Build Options** page.



Important • You cannot create or edit a release in the ThinApp Assistant. If no releases exist, you can simply click the **Build** toolbar button to create a new release or open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build a ThinApp application. For more information, see [Creating and Building Releases](#).

If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.



Task: **To select releases to build:**

1. Open the **Build Options** page.
2. Select the releases in the **Releases** tree that you want to build a ThinApp application for.



Important • When you select a release on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build a ThinApp application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and a ThinApp application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See *Performing Batch Builds*.

Enabling ThinApp Application Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **ThinApp Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build a ThinApp application for this Windows Installer package. To do this, perform the following steps:



Task: **To enable ThinApp application building when in Direct Edit Mode:**

1. Open a Windows Installer package or a transform file in InstallShield. It will be opened in Direct Edit Mode or Direct MST Mode, and the Build function (**Build** on the **Build** menu and the **Build** toolbar button) will be disabled.
2. Open the **Build Options** page of the ThinApp Assistant.
3. Select the **Build ThinApp application** option. After you select this option, the **Build ThinApp application** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.


Including Additional Windows Installer Packages in a ThinApp Application

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual package, you can also use the ThinApp Assistant to convert an application suite of multiple Windows Installer packages into one virtual package.

To include additional Windows Installer packages in a ThinApp application, set the **Would you like to include additional MSI files in the virtual package?** option on the **Build Options** page to **Yes**, and then select the packages that you want to add.



Task: *To include additional Windows installer packages in a ThinApp application:*

1. Open the **Build Options** page.
2. Set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**.
3. Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.
 - The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.
 - Use the Delete button () to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application

You can choose to build a Windows Installer package to assist in the distribution of a ThinApp application. This simplifies the deployment of a ThinApp application by enabling you to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management.

To build a Windows Installer file with your ThinApp application, select the **Generate a Windows Installer (MSI) file as part of the build output** option on the **Build Options** page of the ThinApp Assistant. By default, this option is not selected.

This Windows Installer file can be run to properly install the ThinApp application on an end-user's desktop. A ThinApp application installed using a Windows Installer package can be uninstalled using **Add or Remove Programs** in the Control Panel.

Setting ThinApp Log Monitor Tracing Options

ThinApp Log Monitor is an application in the ThinApp Suite that allows you to record detailed information about any application's execution history for later review. The following events are recorded:

- **API calls**—Win32 API calls with parameter and result information made by applications running in the ThinApp virtual operating system
- **Errors**—A list of potential errors, exceptions, and security events within the application
- **Loaded DLLs**—A list of all DLLs loaded by the application and address ranges

Log Monitor is launched by selecting a shortcut in the ThinApp Suite group on the Windows Start menu.

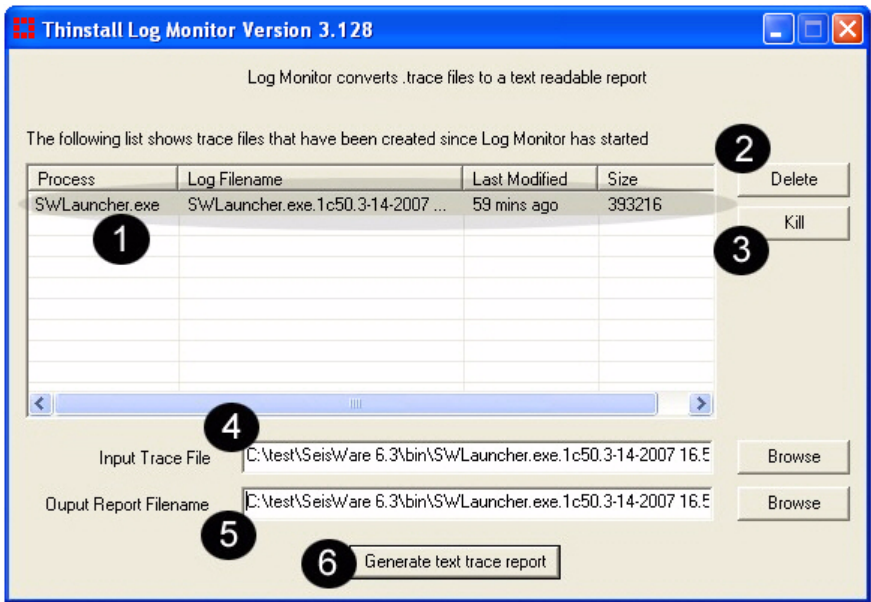



Figure 10-34: ThinApp Log Monitor

Log Monitor displays the following information:

Table 10-30 • ThinApp Log Monitor Interface

#	Name	Description
1	Process List	<p>Any new ThinApp process which has been started after Log Monitor begins will be listed. If you click on one of the processes, the Input Trace File and Output Report Filename fields are automatically populated.</p> <div><p>Note • If the application was built with the Disable Log Monitor Tracing option on the Build Options page selected, it will not be listed.</p></div>
2	Delete	Click to delete trace files for the selected processes in the Process List.
3	Kill	Click to kill currently running process that is selected in the Process List. You would do this to stop a process from logging additional entries once an error condition has been reached.
4	Input Trace File	Click Browse to manually browse for a trace file to convert.
5	Output Report File	The file listed in this field is generated when you click Generate text trace report . This report should be viewed with a text editor that supports UNIX-style line breaks such as Wordpad or Word (not Notepad).

Disabling Log Monitor Tracing

If you do not want to allow ThinApp Log Monitor tracing in a ThinApp application, select the **Disable Log Monitor Tracing** option on the **Build Options** page.



Task: *To disabling the ThinApp Log Monitor tracing capabilities:*

1. Open the **Build Options** page.
2. Select the **Disable Log Monitor Tracing** option.
3. Build the ThinApp application.

Setting AppLink Options



Note • The AppLink Settings feature requires ThinApp 4.x. If you are using Thinstall 3.x, any AppLink settings that you define will be ignored.

The AppLink (Application Link) feature enables you to configure relationships between ThinApp applications that work together. You can set AppLink settings for the current ThinApp application on the **AppLink Settings** dialog box, which is opened by clicking the **AppLink Settings** option in the **More Options** menu of the ThinApp Assistant **Build Options** page.

You can use the AppLink feature to perform the following tasks:

- **Linking runtime components to applications**—You can link runtime components to the applications that use them. For example, you can link a package containing the Java runtime environment (JRE) or ODBC drivers to a package containing a browser application.
- **Linking add-ons and plug-ins to applications**—You can link add-ons and plug-ins to applications. For example, Microsoft Office add-ons can be linked to applications or Adobe Photoshop plug-ins can be linked to a package containing Photoshop.
- **Linking packaged applications to service packs**—You can link packaged applications to service packs. By using AppLink, you can upgrade or roll back your service packs by changing the service pack that you capture and link to its parent application.

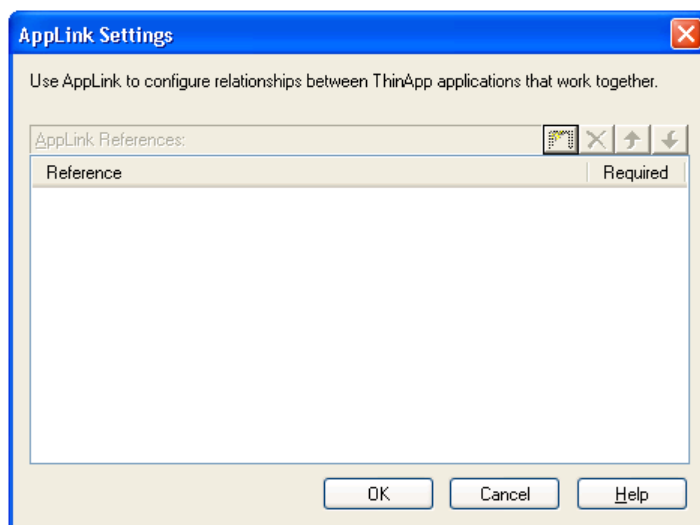
To set AppLink options for a ThinApp application, perform the following steps.



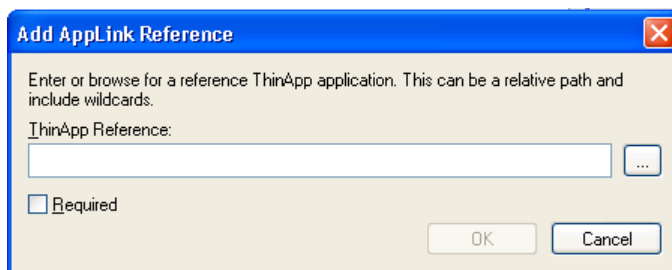
Task:

To configure AppLink settings for your ThinApp application:

1. On the **Build Settings** page of the ThinApp Assistant, click the **AppLink Settings** option in the **More Options** menu. The **AppLink Settings** dialog box opens.



2. Click the Browse button to open the **Add AppLink Reference** dialog box.



3. In the **ThinApp Reference** box, enter the relative (runtime) path to the existing ThinApp application that you want to link to.
 - If you want to add multiple applications, repeat the procedure as necessary.
 - You can also use wild cards. See [Security and Authorization](#)

- The order in which packages are imported can be changed by selecting a package and clicking the up and down arrows. See [Collisions and Order of Import](#) for more information on order.



Note • Required and Optional links are listed on the AppLink Settings dialog box together and the order can be changed using the up and down arrows. However, at runtime, all of the applications in the Required category are read first, before those in the Optional category, even if applications in the Optional category were originally higher in the list. When the AppLink Settings dialog box is reopened, the AppLink References will be grouped by category rather than be in the order that was arranged prior to closing the dialog box. In other words, the category order (Required and Optional) overrides the order set by the user.

- To delete a package you have added, select the package and click the Delete () button.



Important • When linking to a ThinApp application that has only one shortcut, select its .EXE file. When linking to a ThinApp application that has more than one shortcut, select either its Package.DAT file (if the ThinApp application was built with AdminStudio) or its primary executable file (if the ThinApp application was built with ThinApp).



Important • On the **Add AppLink Reference** dialog box, if you click Browse and browse for a ThinApp application, the absolute path to that application is entered, such as `C:\Program Files\AppName\filename.exe`. In that case, the parent ThinApp application needs that linked application to be found at the specified absolute path location at runtime, which is unlikely. Therefore, it is recommended that you enter a relative path name.

4. If you want this package to be required, select the **Required** option. If a required package is missing from the virtual package, it will fail to run. Note the following about required packages:
 - If any specified package fails to import, an error message will be displayed and the parent executable file will exit.
 - If a wildcard pattern is used to specify a package, no error message is displayed if no files match the wildcard pattern. Therefore, if a wildcard pattern is used to specify a package, the reference is always optional.
 - To continue even if load errors occur, make the package references optional instead.
5. Click **OK** to return to the **AppLink References** dialog box. The item you selected is now listed in the **AppLink References** list.
6. Click **OK** to return to the **Build Options** page.

Setting AppSync Options



Note • The AppSync Settings feature requires ThinApp 4.x. If you are using Thinstall 3.x, any AppSync settings that you define will be ignored.

AppSync (Application Sync) enables you to automatically keep deployed ThinApp applications up to date. When an application starts up, AppSync can query a Web server to see if an updated version of the package is available. If an update is available, the differences between the existing package and the new package will be downloaded and used to construct an updated version of the package. The updated package will be used for future deployments.

You can use the AppSync feature to perform the following tasks:

- **Distribute runtime components separately**—You can use AppSync to distribute runtime components separately from the applications that use them. For example, the Java Runtime Environment (JRE) or ODBC drivers.
- **Apply layered service packs to applications**—You can use AppSync to apply layered service packs to your applications. Application Sync enables you to distribute service packs and roll back to previous versions, if necessary.

On the AppSync Settings dialog box, you specify the location of the update, the message displayed to the user, and the expiration settings. You set AppSync settings for the current ThinApp application on the **Build Options** page of the ThinApp Assistant. To configure AppSync settings for a ThinApp application, perform the following steps:



Task: To configure AppSync settings for your ThinApp application,

1. On the **Build Settings** page, click the **AppSync Settings** option in the **More Options** menu. The **General** tab of the **AppSync Settings** dialog box opens.

The screenshot shows the 'AppSync Settings' dialog box with the 'General' tab selected. The dialog has a title bar with a close button. Inside, there are two tabs: 'General' and 'Expiration'. The 'General' tab contains the following fields and controls:

- A text area for 'Url:' with a placeholder text: 'ThinApp uses AppSync to check for application updates. Provide the URL to the web server that will host the update.'
- A text area for 'Message:' with a placeholder text: 'The application has been successfully updated.'
- A dropdown menu for 'Frequency:' currently showing '1d'.
- A checkbox labeled 'Clear Sandbox' which is currently unchecked.
- At the bottom, there are three buttons: 'OK', 'Cancel', and 'Help'.

2. In the **Url** field, specify the location of the Web server that hosts application updates. When entering the URL, consider the following:

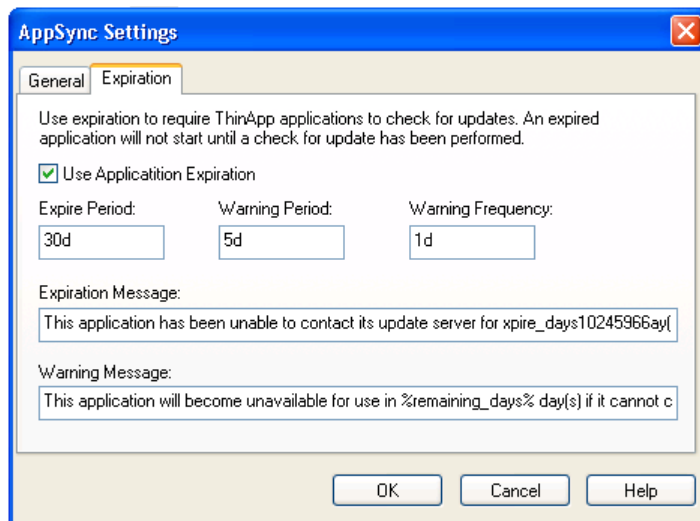
- **Supports HTTP and HTTPS**—Application Sync works over both the HTTP (unsecure) and HTTPS (secure) protocol.
- **Can include login information**—You can include a user name and password in the URL that will be used for basic authentication. The standard Windows/Internet Explorer proxy setting is respected. For example:

`https://www.example.com/some/path/PackageName.exe`

3. In the **Message** field, enter the information you want to display to the user when the ThinApp application is updated. By default, the following is entered:

The application has been successfully updated.

4. By default, a package will connect to the Web server once per day to see if an updated version is available. You can set the frequency by modifying the **Frequency** setting. For example, to set the **Frequency** to 2 days, enter **2d**. For 2 weeks, enter **2w**, etc.
5. If you want to automatically clear the sandbox after an update, select the **Clear Sandbox** option. By default, this option is not selected.
6. Click **Expiration** to open the **Expiration** tab. On this tab, you can specify that a ThinApp application is required to check for updates at a defined frequency. If the ThinApp application fails to successfully check for updates within that defined frequency, it will fail to run.



7. To require that an application has to check for updates at a specified frequency, select the **Use Application Expiration** option

8. In the **Expire Period** box, enter the update frequency in minutes (m), hours (h), or days (d). For example:

- To set the period to 30 days, enter **30d**.
- To set the period to 72 hours, enter **72h**.
- If you do not want the package to expire, clear the **Use Application Expiration** check box.

If the Web server cannot be reached, meaning that the update fails, the package will continue to work until the **Expire Period** is reached. The default setting is 30 days.

9. In the **Warning Period** box, enter the amount of time prior to expiration that the user is first warned. For example, to set the period at 5 days, enter **5d**.
10. In the **Warning Frequency** box, enter the frequency that a warning message will be displayed to the user before the package expires. With the default of one day, the warning message will be displayed once per day only. To configure the warning to pop up on every application launch, enter **0**. To configure it to pop up every 4 days, enter **4d**.

Note the following about warning frequency:

- After the warning period has started, the Web server will be checked on every launch of an application, overriding any previous setting.
 - As long as a package has not expired, this parameter checks for new versions and downloads will occur in the background. The user can continue to use the old version.
 - If the application is terminated by the user before the download is complete, the download will resume when a virtual application is launched again. After the download completes, the new version will be activated on the next launch.
 - When the package has expired, the version check and download will happen in the foreground. A progress bar will be shown during the download phase.
11. Before the expiration limit has been reached and a ThinApp application is started, it will try to connect to the Web server and check for a new version. If the connection fails, a message box will be shown. The default message is:

*This application will become unavailable for use in **Warning_Period** days if it cannot contact its update server. Check your network connection to ensure uninterrupted service*

12. After the expiration limit has been reached and a ThinApp application is started, it will try to connect to the Web server and check for a new version. If the connection fails, the message entered in the **Expiration Message** box will be shown and execution will be terminated. The default message is:

*This application has been unable to contact its update server for **Expire_Period** days, so it is unavailable for use. Check your network connection and try again.*



Note • If you use AppSync, VMware recommends that you disable automatic application updates that are configured in your virtual application. Conflicts might occur between the linked packages and the software that is automatically updated. If an automatic update feature updates an application, it stores the updates in the sandbox. If AppSync then updates the application to a different version, the updates stored in the sandbox take precedence over the files contained in the version that AppSync created. The order of precedence for the update files are those in the sandbox, then the virtual operating system, and then the physical machine.

Building a ThinApp Application

The method for building a ThinApp application depends upon what file you have open—an InstallShield project or a Windows Installer package.

- [Building a ThinApp Application for an InstallShield Project](#)
- [Building a ThinApp Application for a Windows Installer Package](#)

Building a ThinApp Application for an InstallShield Project

To build a ThinApp application for an InstallShield project, perform the following steps:



Task:

To build a ThinApp application for an InstallShield project:

1. Open the InstallShield project in InstallShield.
2. On the **Releases** view of the Installation Designer, make sure that at least one release has been created, and select the release that you want to build.



Important • You cannot create or edit a release in the ThinApp Assistant. If no releases exist, or if you want to create a new release, open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build a ThinApp application. For more information, see [Creating and Building Releases](#).

3. Open the **Build Options** page of the ThinApp Assistant.
4. In the **Releases** tree, select the same release that is selected on the **Releases** view of the InstallShield Installation Designer. This is the release that you will build a ThinApp application for.



Important • When you select a release on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build a ThinApp application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what was selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and a ThinApp application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.

To build more than one release at a time, perform a batch build. See *Performing Batch Builds*.

5. Click the **Build** toolbar button (or select **Build Release** on the **Build** menu) to start building the active release.

The output of the build will be a Windows Installer package and a ThinApp application. For information on the files included in a ThinApp application, see [Components of a ThinApp Application](#).

Building a ThinApp Application for a Windows Installer Package

To build a ThinApp application for a Windows Installer package, perform the following steps:



Task:

To build a ThinApp application for a Windows Installer package:

1. Do one of the following to open a Windows Installer package:
 - On the **File** menu, select **Open** and select a Windows Installer package (.msi).
 - On the **File** menu, select **Open** and select a transform file (.mst). The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
 - On the **File** menu, select **New** to open the **New Project** dialog box. Select **Transform** and click **OK**. The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
2. Use the Installation Designer to make any desired edits to the Windows Installer package or Transform file, and use the ThinApp Assistant to set ThinApp application options.
3. Save the edits to the Windows Installer package or transform file by selecting **Save** on the **File** menu.
4. On the **Build Options** page of the ThinApp Assistant, select the **Build ThinApp application** option. The **Build Virtual Package** button is enabled.
5. Click the **Build Virtual Package** button to start building the ThinApp application.

The output of the build will be a ThinApp application. For information on the files included in a ThinApp application, see [Components of a ThinApp Application](#).



Note • For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see *Virtualization Conversion Errors and Warnings*.

ThinApp Assistant Reference

Reference information about the ThinApp Assistant is organized into the following sections:

- [Pages](#)
- [Dialog Boxes](#)
- [Building ThinApp Applications Using the Command Line](#)
- [ThinApp Application Conversion Error and Warning Messages](#)
- [Application Features Requiring Pre- or Post-Conversion Actions](#)

Pages

The ThinApp Assistant is comprised of the following pages:

- [ThinApp Assistant Home Page](#)
- [General Settings Page](#)
- [Files & Folders Page](#)
- [Applications Page](#)
- [Registry Page](#)
- [Build Options Page](#)

ThinApp Assistant Home Page

The ThinApp Assistant Home page displays a diagram that illustrates the process of creating a ThinApp application.

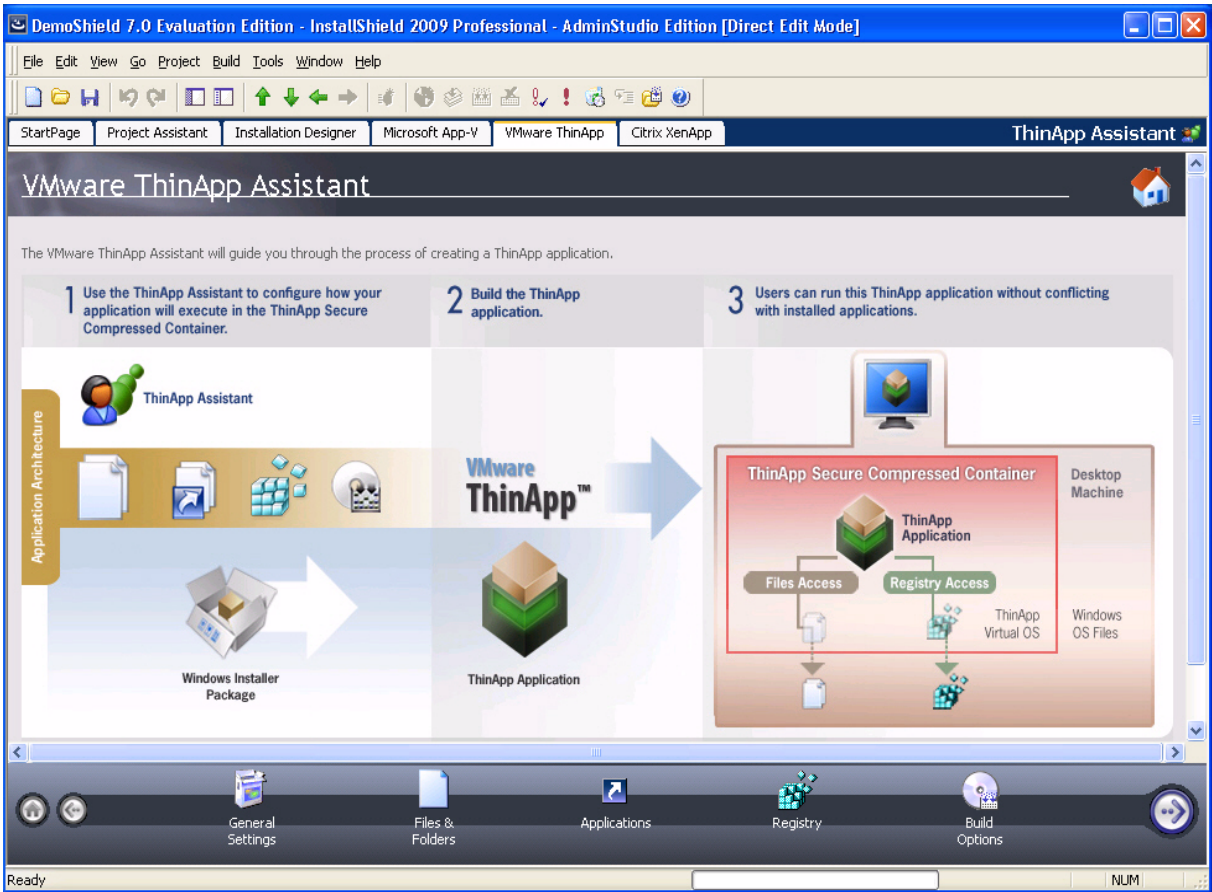








Figure 10-35: ThinApp Assistant Home Page

Click the following icons in the navigation bar at the bottom of the page to navigate through the ThinApp Assistant interface:

Table 10-31 • Navigation Bar Icons

Icon	Destination
	General Settings Page
	Files & Folders Page

Table 10-31 • Navigation Bar Icons

Icon	Destination
	Applications Page
	Registry Page
	Build Options Page
	Go to next page.
	Jump back to previous page.
	ThinApp Assistant Home Page

General Settings Page

On the **General Settings** page in the **ThinApp Assistant**, you specify Sandbox options, including options to control access to the ThinApp application using Active Directory.

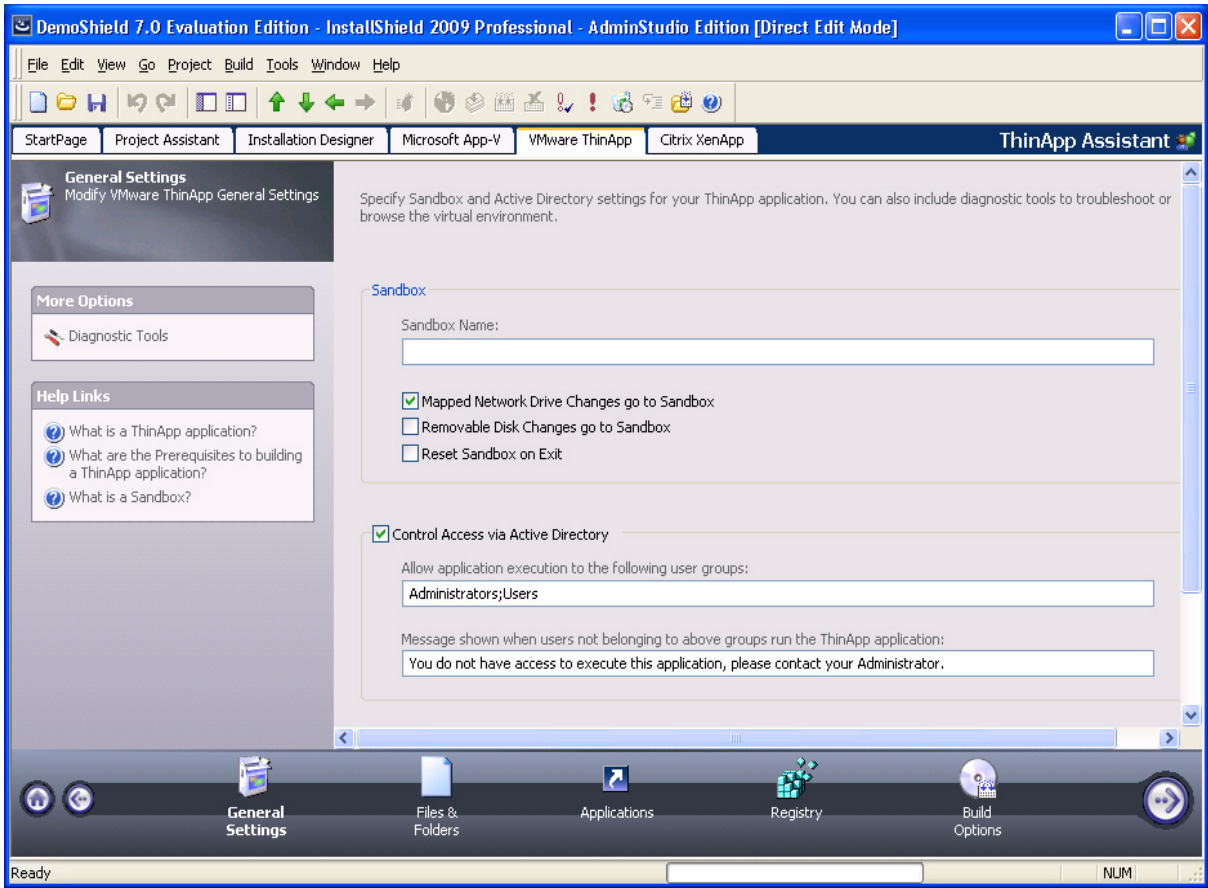




Figure 10-36: ThinApp Assistant General Settings Page

The **General Settings** page includes the following options:

Table 10-32 • General Settings Page

Option	Description
Sandbox Name	When a ThinApp application is built, a Sandbox cache is created in the following location: c:\Documents & Settings\USER_NAME\Application Data\ThinApp\SANDBOX_NAME By default, AdminStudio names the Sandbox by assigning it a unique GUID. However, if you want to override this default Sandbox name, you may (optionally) enter a new name in the Sandbox Name field.

Table 10-32 • General Settings Page (cont.)

Option	Description
Mapped Network Drive Changes go to Sandbox	Enable this option if you want changes for Network mapped drives to be saved in the sandbox. By default, users can read and write normally to network mapped drives.
Removable Disk Changes go to Sandbox	Enable this option if you want changes for removable disks to be saved in the sandbox. By default users can read and write normally to removable disks.
Reset Sandbox on Exit	Select this option to delete the sandbox content when the application exits. This resets the ThinApp application to its original captured state.
Control Access via Active Directory	<p>If you want to control the access of users to a ThinApp application by specifying Active Directory groups, select this option and enter the names of those groups.</p> <p>At build-time, ThinApp would then assign a unique GUID-like number to uniquely identify each Active Directory Group that you have identified. Members of those groups will have access to the ThinApp application.</p> <ul style="list-style-type: none"> • Allow application execution to the following user groups—Enter the names of all of the Active Directory groups that you want to have permission to run this ThinApp application, separated by semi-colons, such as: <code>GroupOne;GroupTwo;GroupThree</code> • Message shown when users not belonging to above groups run the ThinApp application—Enter the message that will be displayed when users that do not belong to the specified groups attempt to launch a ThinApp application. <div>  <p>Caution • If you do not select the Control Access via Active Directory option, anyone who has access to a directory containing a ThinApp application will be able to run the application.</p> </div> <div>  <p>Note • For more information, see About Controlling Access to ThinApp Applications.</p> </div>

For testing purposes, you can also choose to include diagnostic tools in your ThinApp application by selecting the **Diagnostic Tools** link in the **More Options** list. For more information, see [ThinApp Diagnostic Tools Dialog Box](#).

Files & Folders Page

On the **Files & Folders** page of the ThinApp Assistant, you can perform the following tasks:

- [View Files and Folders](#)
- [Add Files and Folders](#)
- [Delete Files and Folders](#)
- [Set Isolation Options](#)
- [Modifying the Display of Predefined Folders](#)

View Files and Folders

On the **Files & Folders** page, you can view all of the files and folders that are currently in your ThinApp application.

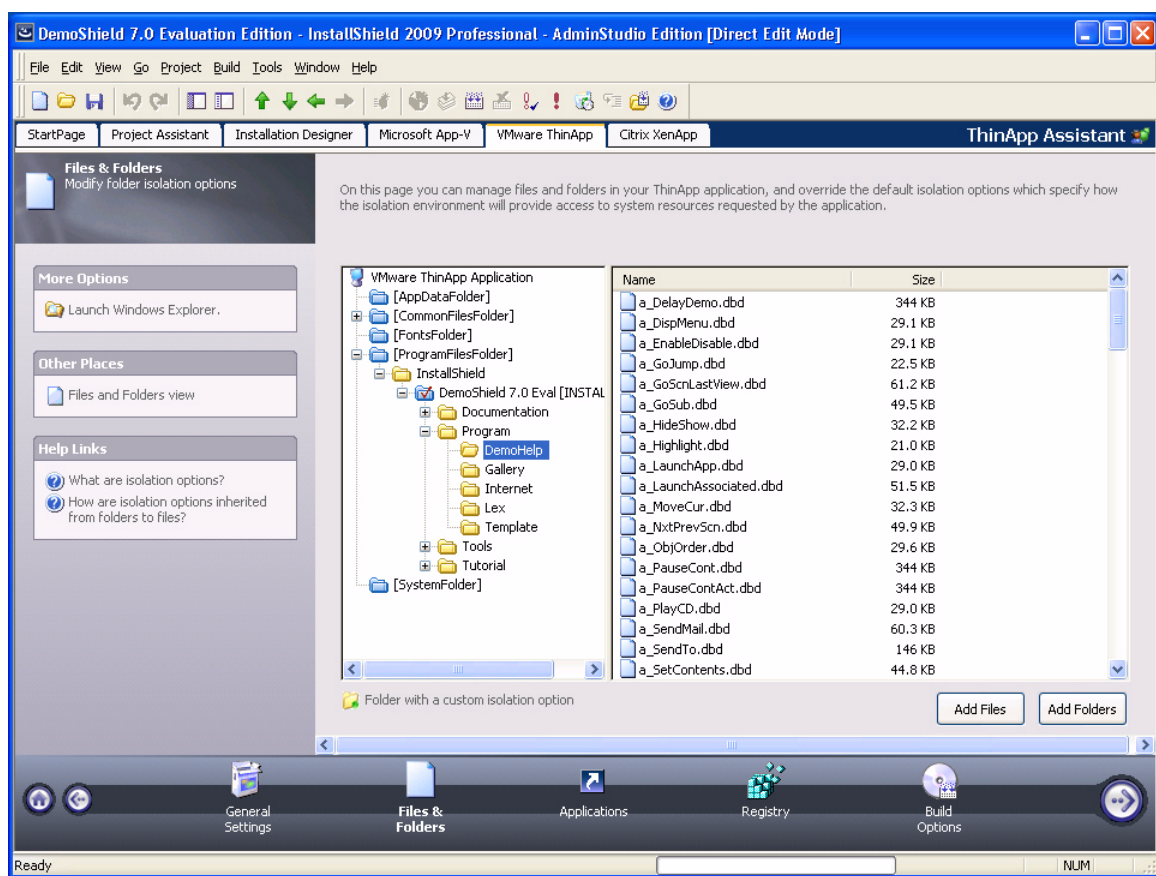


Figure 10-37: ThinApp Assistant Files & Folders Page

Folders are listed in the **VMware ThinApp Application** tree on the left, and all of the files in the selected folder are listed on the right.

- The directories in the tree represent how your application will be organized within its secure compressed container.
- Blue folders are the supported MSI standard folders.
- The folder with the check mark is **INSTALLDIR**, which represents the main product installation directory.

Add Files and Folders

On the **Files & Folders** page, you can use the **Add Files** and **Add Folders** buttons to add new files and folders to include in the ThinApp application. See [Adding, Deleting, and Moving Files and Folders in a ThinApp Application](#).

If you are editing an InstallShield project (not a Windows Installer package), and you are adding a folder to this ThinApp application, you are prompted to choose whether you want to create a dynamic file link to the source folder.

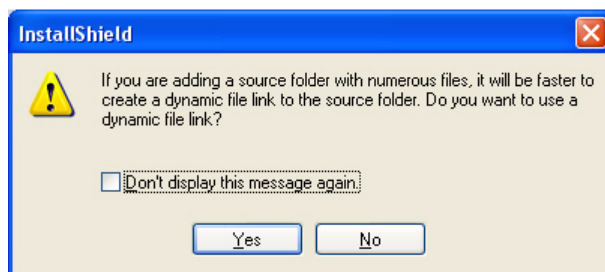


Figure 10-38: Dynamic File Link Option Dialog Box

Indicate whether you want to create a dynamic file link by selecting one of the following:

- **No**—For more flexibility with ThinApp options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
- **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See [Dynamic File Link Settings Dialog Box](#).

Delete Files and Folders

You can delete files and folders from the ThinApp application by selecting the file or folder you want to delete, and selecting **Delete** from the context menu. For more information, see [Deleting Files and Folders](#).



Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Set Isolation Options

ThinApp uses a sandbox virtual environment to control application compatibility and accessibility. The isolation option that is assigned to a folder or registry key specifies how the virtual environment will provide access to system resources requested by the application.

The default settings for isolation options are built into the ThinApp Assistant, and those defaults are adequate for most environments. However, you can override the default settings for selected files, folders, or registry keys to exert control over application interactions with client operating system resources.

You set isolation options by selecting a file or folder and then selecting **Isolation Options** from the context menu. For an overview of the available isolation options, and for instructions on how to set them, see [Setting ThinApp Isolation Options](#).

Modifying the Display of Predefined Folders

You can specify which of the Windows Installer predefined folders are listed in the **VMware ThinApp Application** tree. See [Controlling the Display of Predefined Folders](#).

Applications Page

You define shortcuts to enable users to launch a ThinApp application from within the sandbox virtual environment.

By default, the **ThinApp Assistant** creates ThinApp applications for all of the executable shortcuts that exist in your project. The project's shortcuts are listed in a checklist on the **Applications** page.

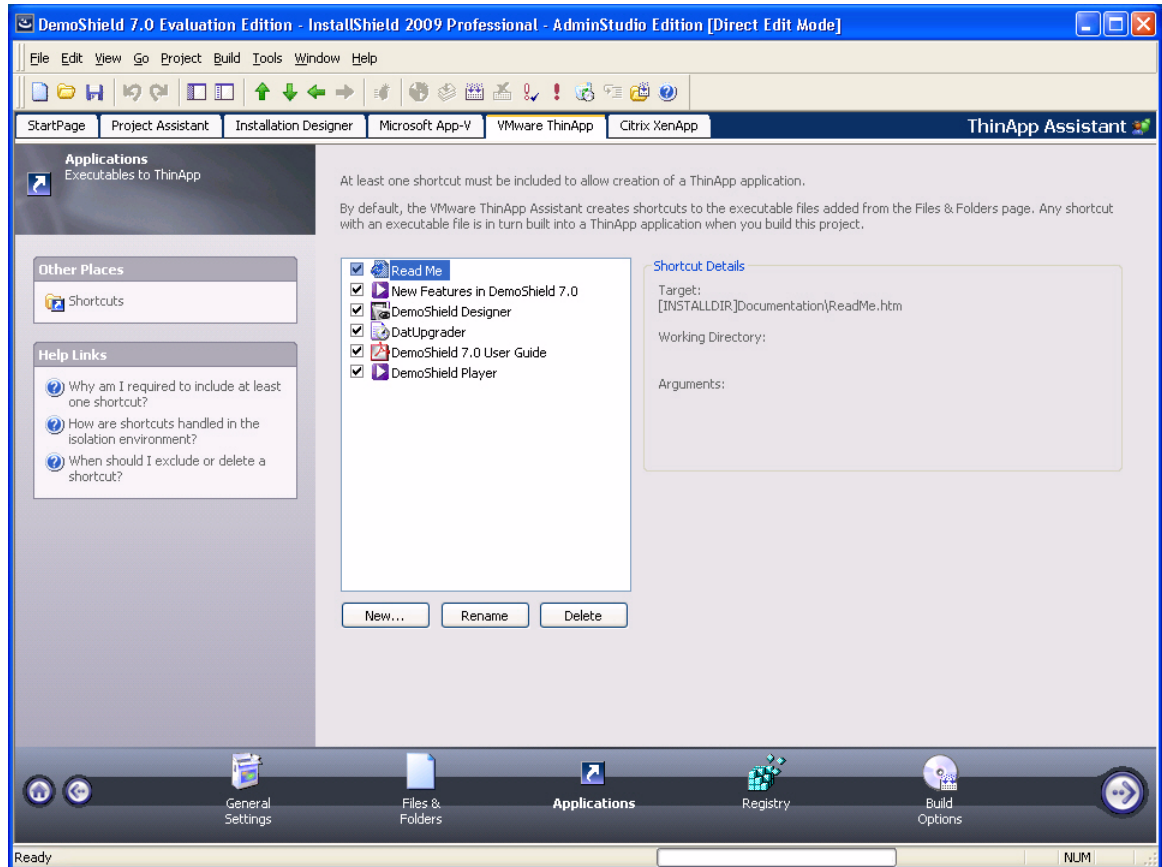


Figure 10-39: ThinApp Assistant Applications Page

Shortcut Requirements

For each ThinApp application, you are required to define **at least one** shortcut. You define application shortcuts to enable users to launch a ThinApp application from within the virtual environment. By default, the ThinApp Assistant creates ThinApp applications for all of the executable shortcuts that exist in your project (or Windows Installer package).

If you build a ThinApp application that does not contain any shortcuts, users will not be able to launch the application.

Difference Between Deleting and Excluding a Shortcut

To prevent a shortcut from being created in the ThinApp application, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the ThinApp application, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding a ThinApp Application](#).
- **Deleting a Shortcut**—When you delete a shortcut, it is removed from both the ThinApp application and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting a ThinApp Application](#).

Managing Shortcuts

On the **Applications** page, you can create, delete, include, exclude, or rename a ThinApp application. For step-by-step instructions, see the following topics:

- [Creating a New ThinApp Application](#)
- [Including an Existing ThinApp Application](#)
- [Excluding or Deleting an Existing ThinApp Application](#)
- [Renaming a ThinApp Application](#)

Registry Page

On the **Registry** page, you can view existing registry keys, values, and data, and add or delete registry items. You can also override the default isolation options for a registry key. Isolation options specify how the virtual environment will provide access to system resources requested by the application.

The default settings for isolation options are built into the ThinApp Assistant, and those defaults are adequate for most environments. However, you can override the default settings for selected registry keys to exert control over application interactions with client operating system resources.

You set isolation options by selecting a registry key and then selecting **Isolation Options** from the context menu. For an overview of the available isolation options, and for instructions on how to set them, see [Setting ThinApp Isolation Options](#).

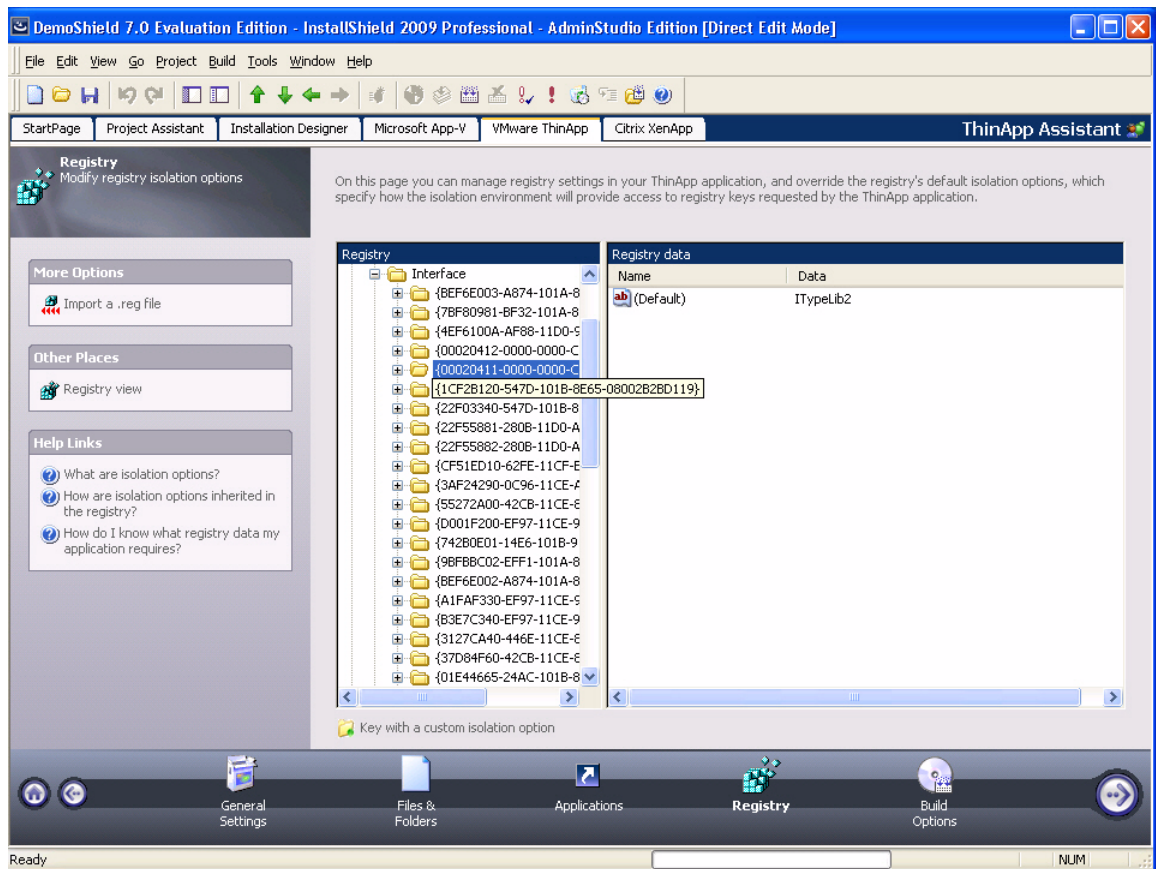


Figure 10-40: ThinApp Assistant Registry Page

Registry items that are listed on this page will be included in the ThinApp application, and those that you delete will not. By default, all new registry keys are isolated.



Tip • To import an existing registry (.reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.



Note • You cannot set isolation options on root registry keys.

Editing the registry on the **Registry** page is performed much like it is performed on the InstallShield **Registry View**. See [Editing the Registry](#).

For information on how to override a registry key's default isolation options, see [Modifying ThinApp Application Registry Settings](#).



Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

Build Options Page

On the **Build Options** page, you can perform the following tasks:

- [Specifying Build Options](#)
- [Including Additional Windows Installer Packages in a ThinApp Application](#)
- [Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application](#)
- [Selecting Releases to Build](#)
- [Enabling ThinApp Application Building When in Direct Edit Mode](#)
- [Clearing the ThinApp Cache](#)
- [Opening the ThinApp Application Folder](#)
- [Building a ThinApp Application](#)
- [Supporting AppSync and AppLink](#)

The options on the Build Options page vary depending upon whether you are editing an InstallShield project or a Windows Installer package:

InstallShield Project

When you open an InstallShield project in InstallShield:

- The **Build Options** page includes a releases tree, and you select the release that you want to build.
- To build the ThinApp application, you click the **Build** button on the toolbar.

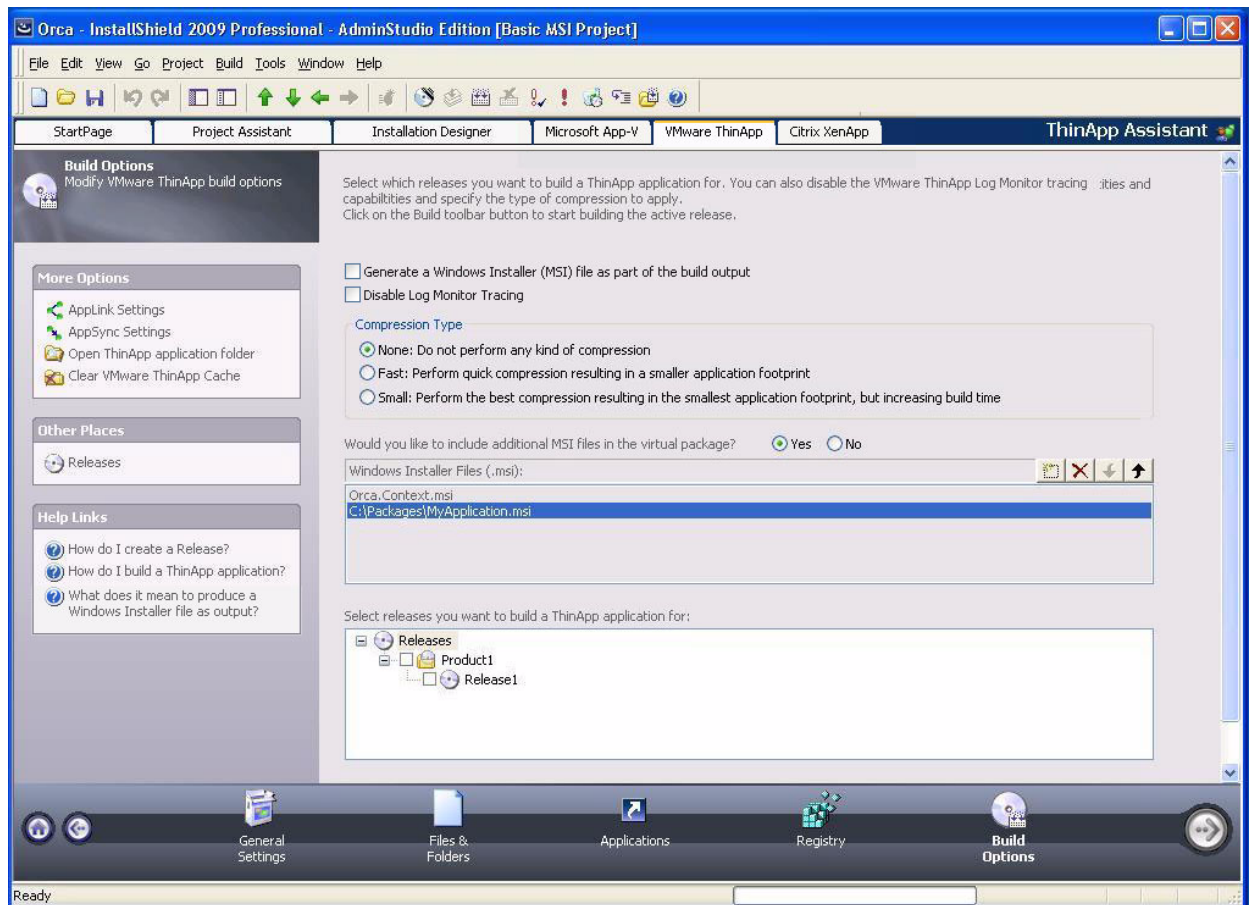


Figure 10-41: Build Settings Page, When in Basic MSI Project Mode

Windows Installer Package [Direct Edit Mode]

When you open a Windows Installer package in InstallShield:

- Because you do not have to select a release for a Windows Installer package, there is no releases tree.
- Because a Windows Installer package has already been built, InstallShield's standard build functionality is disabled. To build the ThinApp application, select the **Build ThinApp application** option and click the **Build Virtual Package** button.

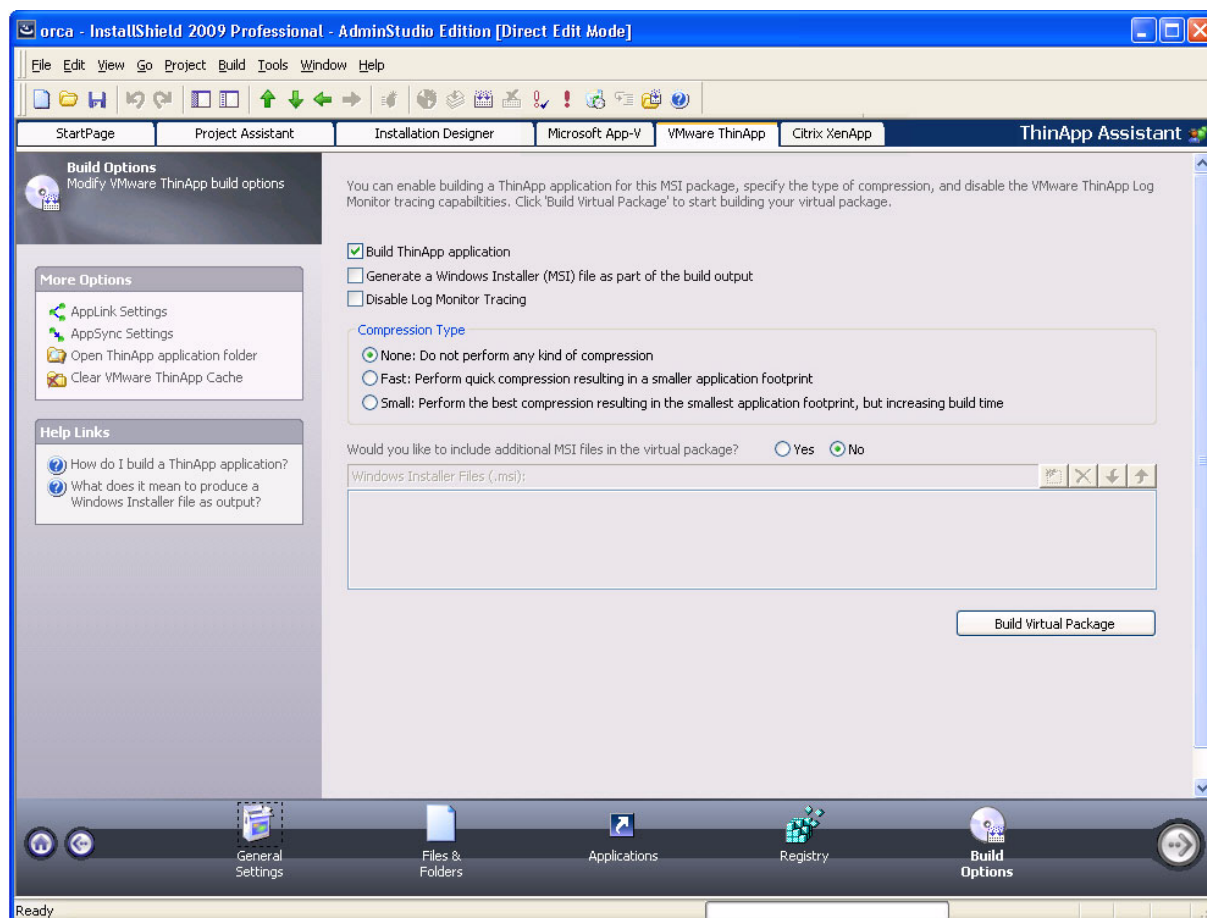


Figure 10-42: Build Settings Page, When in Direct Edit Mode

Specifying Build Options

On the **Build Options** page, you can specify the following options:

Table 10-33 • ThinApp Application Build Options



Option	Description
Build ThinApp Application	<p>(Direct Edit/Direct MST Modes Only) When you directly edit a Windows Installer package, it is not necessary to build the package, because it is already built. Therefore, InstallShield's Build function is disabled. Select the Build ThinApp Application option to enable the Build function. When this option is selected, the Build Virtual Package button is enabled.</p> <p>For more information, see Enabling ThinApp Application Building When in Direct Edit Mode.</p>
Build Virtual Package	<p>(Direct Edit/Direct MST Modes Only) When you directly edit a Windows Installer package, if you select the Build ThinApp Application option, this button is enabled. Click it to build the ThinApp application.</p>
Generate a Windows Installer (MSI) file as part of the build output	<p>You can choose to build a Windows Installer package with your ThinApp application. This enables you to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management to distribute your ThinApp application.</p> <p>To build a Windows Installer file with your ThinApp application, select this option. By default, this option is not selected.</p> <p>For more information, see Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application.</p>
Disable Log Monitor Tracing	<p>Select this option if you do not want to allow ThinApp Log Monitor tracing for a ThinApp application.</p> <p>ThinApp Log Monitor is an application in the ThinApp Suite that allows you to record detailed information about any application's execution history for later review.</p> <p>For more information, see Setting ThinApp Log Monitor Tracing Options.</p>
Compression Type	<p>Select one of the following options to specify the ThinApp application's compression type:</p> <ul style="list-style-type: none"> • None: Do not perform any type of compression • Fast: Perform quick compression resulting in a smaller application footprint • Small: Perform the best compression resulting in the smallest application footprint, but increasing build time.  <p>Note • For more information, see Compressing a ThinApp Application.</p>





Table 10-33 • ThinApp Application Build Options

Option	Description
Would you like to include additional MSI files in the virtual package?	<p>Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. To include additional Windows Installer packages in a ThinApp application, set this option to Yes, and then select the packages that you want to add.</p>  <p>Note • For more information, see Including Additional Windows Installer Packages in a ThinApp Application.</p>

Including Additional Windows Installer Packages in a ThinApp Application

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual package, you can also use the ThinApp Assistant to convert an application suite of multiple Windows Installer packages into one virtual package.

To include additional Windows Installer packages in a ThinApp application, set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**, and then select the packages that you want to add.

- Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.
- The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.
- Use the Delete button () to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Building a Windows Installer Package to Assist in the Distribution of a ThinApp Application

You can choose to build a Windows Installer package to assist in the distribution of a ThinApp application by selecting the **Generate a Windows Installer (MSI) file as part of the build output** option on the **Build Options** page. By default, this option is not selected.

The Windows Installer file can be run to properly install the ThinApp application on an end-user's desktop. This simplifies the deployment of a ThinApp application by enabling you to use enterprise distribution tools such as Microsoft System Center Configuration Manager or Novell ZENworks Configuration Management.

A ThinApp application installed using a Windows Installer package can be uninstalled using **Add or Remove Programs** in the Control Panel.

Selecting Releases to Build

You select the releases that you want to build a ThinApp application for on the **Releases** tree of the **Build Options** page. By selecting a release, you are specifying that whenever that particular release is built, a ThinApp application will also be built.



Note • If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.

About Building Releases

When you select a release on the Releases tree on the **Build Options** page, you are specifying that whenever you build that particular release, you want to also build a ThinApp application for that release. However, the releases that are selected on the **Build Options** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Options** page:

- **Active release selected**—A Windows Installer package and a ThinApp application would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See *Performing Batch Builds*.

About Creating Releases

You cannot create or edit a release in the ThinApp Assistant. If no releases exist, you can simply click the **Build** toolbar button to create a new release or open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build a ThinApp application. For more information, see [Creating and Building Releases](#).

If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Options** page is not displayed.

Enabling ThinApp Application Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **ThinApp Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build a ThinApp application for this Windows Installer package. To enable the **Build** button to build just the ThinApp application, select the **Build ThinApp application** option on the **Build Options** page.

After you select this option, the **Build ThinApp application** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.

Clearing the ThinApp Cache

When you perform compressed builds, large temporary files are saved in a cache location. To delete all of these temporary files, select the **Clear the VMware ThinApp Cache** option in the **More Options** list on the **Build Options** page

Opening the ThinApp Application Folder

To quickly open the folder containing the ThinApp application files that were generated when this InstallShield project or Windows Installer package was built, click **Open ThinApp application folder** in the **More Options** menu.

Building a ThinApp Application

The method for building a ThinApp application depends upon what file you have open—an InstallShield project or a Windows Installer package. For detailed instructions, see one of the following topics:

- [Building a ThinApp Application for an InstallShield Project](#)
- [Building a ThinApp Application for a Windows Installer Package](#)

Supporting AppSync and AppLink

To configure AppSync and AppLink settings for your ThinApp application, click the **AppSync Settings** or **AppLink Settings** option in the **More Options** menu. For more information, see the [AppSync Settings Dialog Box](#) or the [AppLink Settings Dialog Box](#).

Dialog Boxes

The ThinApp Assistant includes the following dialog boxes:

- [ThinApp Diagnostic Tools Dialog Box](#)
- [Folder Isolation Options Dialog Box](#)
- [Registry Isolation Options Dialog Box](#)
- [AppLink Settings Dialog Box](#)
- [Add AppLink Reference Dialog Box](#)
- [AppSync Settings Dialog Box](#)

ThinApp Diagnostic Tools Dialog Box

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **General Settings** page, you can choose to include the Windows Command Prompt and Registry Editor diagnostic tools with your ThinApp application.

If you include diagnostic tools with your ThinApp application, you will be able to look at the registry or file system for the application while it is running in its virtual environment. For example, if you were running a ThinApp application and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.



Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the ThinApp application. However, these tools may not be compatible with other operating systems.

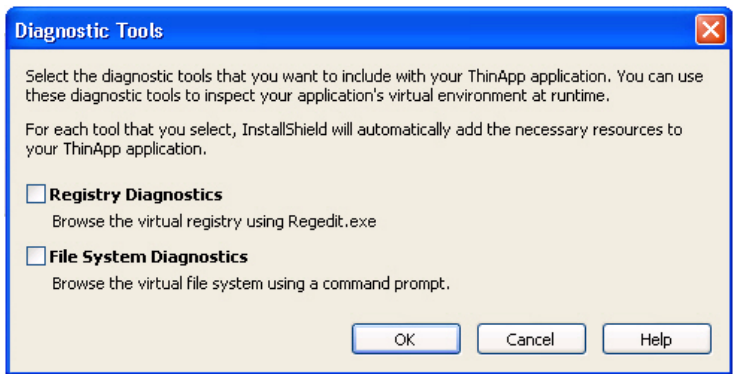


Figure 10-43: Diagnostic Tools Dialog Box

You can use these diagnostic tools to inspect your application’s virtual environment at runtime. You have the following options:

Table 10-34 • Diagnostic Tools Dialog Box Options

Option	Description
Registry Diagnostics	Select this option if you want to include <code>regedit.exe</code> with your ThinApp application so that you can browse the registry.
File System Diagnostics	Select this option if you want to be able to browse the ThinApp application’s virtual environment file system using a command prompt.

Launching the Diagnostic Tools Within the Virtual Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the ThinApp application.

When the user runs this ThinApp application, two additional shortcuts will be available in the application’s shortcut folder: The names of these shortcuts will reflect the application name, such as:

[ProductName] Registry
[ProductName] File System

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application’s virtual environment.

Folder Isolation Options Dialog Box

On the **Folder Isolation Options** dialog box, you can override the default isolation options for the selected folder.

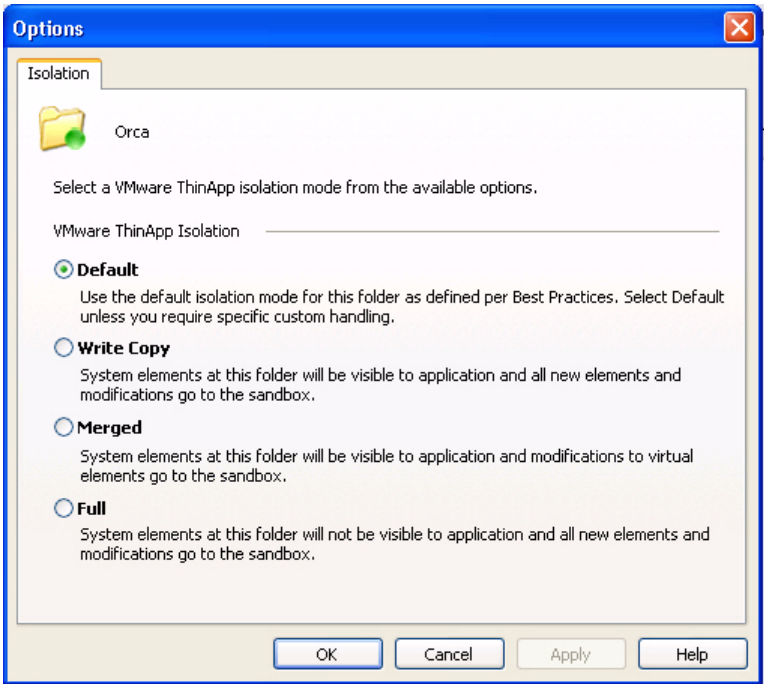


Figure 10-44: Folder Isolation Options Dialog Box



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, ThinApp, and registry settings.

The **Folder Isolation Options** dialog box includes the following options:

Table 10-35 • ThinApp Isolation Options

Option	Visibility of System Elements	Modifications to Virtual Elements	Modifications to System Elements	New Elements	If System and Virtual Element at Same Location
Default	As defined internally by the ThinApp Assistant				
Write Copy	Visible	Sandbox	Sandbox	Created in Sandbox	Sees Virtual Element
Merged	Visible	Sandbox	System	Created in System	Sees Virtual Element

Table 10-35 • ThinApp Isolation Options

Option	Visibility of System Elements	Modifications to Virtual Elements	Modifications to System Elements	New Elements	If System and Virtual Element at Same Location
Full	Not Visible	Sandbox	N/A (System elements cannot be modified)	Created in Sandbox	N/A (System elements cannot be read)

ThinApp Isolation Option Use Scenarios

The following table describes scenarios where you would use each isolation option:

Table 10-36 • Use Scenarios for ThinApp Isolation Options

Option	Use Scenario
Write Copy	<p>You would use Write Copy isolation when:</p> <ul style="list-style-type: none"> Application was not designed or tested for multi-user environments and expects it can modify files and keys without impacting other users. Application expects write permission to Global locations and was not designed for locked-down desktop environments found in corporate environments or Windows Vista. <p>With Write Copy isolation, ThinApp makes copies of registry keys and files written by the application and performs all of the modifications in a user-specific sandbox. With this type of isolation, the ThinApp applications believe that they have global write permissions, while they really only modify the sandbox directory.</p>
Merged	<p>You would use Merged isolation when the ThinApp application needs write access to user-specific storage areas, like the Desktop and My Documents.</p>
Full	<p>You would use Full isolation when a ThinApp application needs to run on a machine where earlier or later versions of the same application are either installed or were not uninstalled correctly.</p> <p>For directories and registry keys that have Full isolation, the ThinApp application will not be aware of any host computer file that might exist, and it sees only virtual files and registry keys at fully isolated locations.</p>

Registry Isolation Options Dialog Box

On the **Registry Isolation Options** dialog box, you can override the default isolation options for the selected registry key.

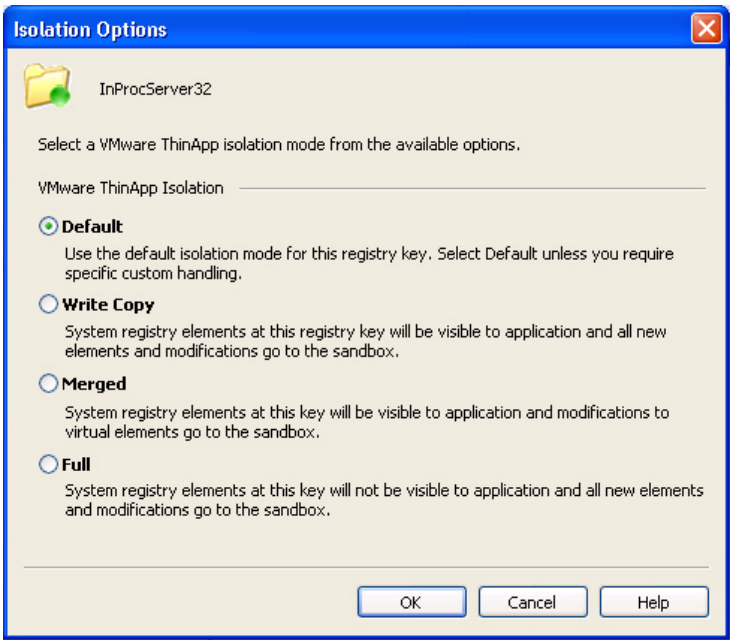


Figure 10-45: Registry Isolation Options Dialog Box



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects, ThinApp, and registry settings.

The **Registry Isolation Options** dialog box includes the following options:

Table 10-37 • ThinApp Isolation Options

Option	Visibility of System Elements	Modifications to Virtual Elements	Modifications to System Elements	New Elements	If System and Virtual Element at Same Location
Default	As defined internally by the ThinApp Assistant				
Write Copy	Visible	Sandbox	Sandbox	Created in Sandbox	Sees Virtual Element
Merged	Visible	Sandbox	System	Created in System	Sees Virtual Element
Full	Not Visible	Sandbox	N/A (System elements cannot be modified)	Created in Sandbox	N/A (System elements cannot be read)

ThinApp Isolation Option Use Scenarios

The following table describes scenarios where you would use each isolation option:

Table 10-38 • Use Scenarios for ThinApp Isolation Options

Option	Use Scenario
Write Copy	<p>You would use Write Copy isolation when:</p> <ul style="list-style-type: none"> • Application was not designed or tested for multi-user environments and expects it can modify files and keys without impacting other users. • Application expects write permission to Global locations and was not designed for locked-down desktop environments found in corporate environments or Windows Vista. <p>With Write Copy isolation, ThinApp makes copies of registry keys and files written by the application and performs all of the modifications in a user-specific sandbox. With this type of isolation, the ThinApp applications believe that they have global write permissions, while they really only modify the sandbox directory.</p>
Merged	<p>You would use Merged isolation when the ThinApp application needs write access to user-specific storage areas, like the Desktop and My Documents.</p>
Full	<p>You would use Full isolation when a ThinApp application needs to run on a machine where earlier or later versions of the same application are either installed or were not uninstalled correctly.</p> <p>For directories and registry keys that have Full isolation, the ThinApp application will not be aware of any host computer file that might exist, and it sees only virtual files and registry keys at fully isolated locations.</p>

AppLink Settings Dialog Box



Note • The AppLink Settings feature requires ThinApp 4.x. If you are using Thinstall 3.x, any AppLink settings that you define will be ignored.

The AppLink (Application Link) feature enables you to configure relationships between ThinApp applications that work together. You can set AppLink settings for the current ThinApp application on the **AppLink Settings** dialog box, which is opened by clicking the **AppLink Settings** option in the **More Options** menu of the ThinApp Assistant **Build Options** page.

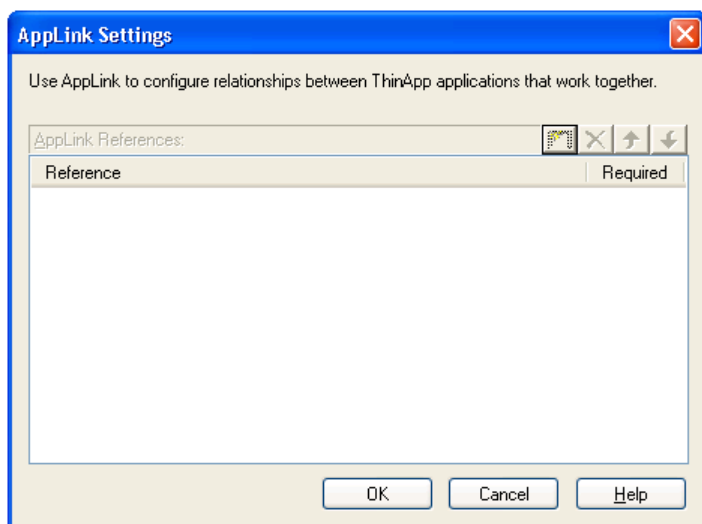


Figure 10-46: AppLink Settings Dialog Box

You can use the AppLink feature to perform the following tasks:


- **Linking runtime components to applications**—You can link runtime components to the applications that use them. For example, you can link a package containing the Java runtime environment (JRE) or ODBC drivers to a package containing a browser application.
- **Linking add-ons and plug-ins to applications**—You can link add-ons and plug-ins to applications. For example, Microsoft Office add-ons can be linked to applications or Adobe Photoshop plug-ins can be linked to a package containing Photoshop.
- **Linking packaged applications to service packs**—You can link packaged applications to service packs. By using AppLink, you can upgrade or roll back your service packs by changing the service pack that you capture and link to its parent application.

The **AppLink Settings** dialog box has the following options:

Table 10-39 • AppLink Settings Dialog Box

Option	Description
AppLink References	<p>List of ThinApp applications that are linked to the open ThinApp application. The following information is listed:</p> <ul style="list-style-type: none"> • Reference—List of linked ThinApp applications, including the application location and name. • Required—If Yes is listed in this column, the linked application must be available in order for the ThinApp application to run. If the linked application cannot be found, the ThinApp application will fail to run. See Required and Optional Linked Applications for more information.

Table 10-39 • AppLink Settings Dialog Box

Option	Description
Browse Button	Click the Browse button to open the Add AppLink Reference dialog box, where you can add a linked application to the AppLink Reference list. For more information, see Add AppLink Reference Dialog Box .
Up and Down Arrows	<p>ThinApp uses a “last import wins” policy to determine what happens when two packages are imported that have the same files or registry keys. Therefore, you can use the Up and Down arrows to order the list of linked applications. See Collisions and Order of Import for more information.</p>  <p>Note • Initially, the Required and Optional linked applications are listed on this dialog box together, and you can change the order of these applications using the Up and Down arrows. However, at runtime, the linked applications in the Required category are read first, before those in the Optional category, even though an Optional application might have been listed before a Required application in the AppLink References list. Also, each time the AppLink Settings dialog box is reopened, the Required linked applications will be grouped at the top of the list, before all Optional applications.</p>

Required and Optional Linked Applications

When an application is linked to a ThinApp application, it can be designated to be either Required or Optional:

Required Applications

If a package is required, it has a mark in the **Required** column. If this package is missing from the virtual package, it will fail to run.

- If any specified package fails to import, an error message will be displayed and the parent executable file will exit.
- If a wildcard pattern is used to specify a package, no error message is displayed if no files match the wildcard pattern. Therefore, if a wildcard pattern is used to specify a package, the reference is always optional.
- To continue even if load errors occur, make the package references optional instead.

Optional Applications

If a package does not have a mark in the **Required** column, it is optional. An optional package operates the same as a required package except that if an import fails to load, the error is ignored and the main application will start executing.

Collisions and Order of Import

ThinApp uses a “last import wins” policy to determine what happens when two packages are imported that have the same files or registry keys.

For example, if PackageA.exe has c:\myinfo.txt in its virtual file system and PackageB.exe also has c:\myinfo.txt in its virtual file system, ThinApp will determine what happens based on which package is imported last.

- **Package order in the AppLink References list**—If PackageA.exe is listed before PackageB.exe on the AppLink References list, PackageB.exe's copy of c:\myinfo.txt will be used. But if PackageB.exe is listed before PackageA.exe on the AppLink References list, PackageA.exe's copy of c:\myinfo.txt will be used.
- **Wild cards**—When wild cards are used, alphabetical order is used to load packages, so if you enter Package*.exe in the AppLink References list, PackageB.exe will be loaded last (after PackageA.exe), so its copy of c:\myinfo.txt will be used.
- **VB scripts**—If two or more packages include VB scripts, the order of execution for the VB Scripts will be alphabetical order by the name of the package. If two packages contain a VB script with the same name, the “last import wins” policy will be used to execute only the version of the VB script from the last imported package containing a script with that name.



Caution • Because VB Script name collisions could cause scripts from other packages not to be executed, it is important to use unique name for VB Script filenames.

Security and Authorization

The user running the ThinApp application must be a member of all PermittedGroups sections for all of the linked (imported) ThinApp applications. If this is not the case, an **Access Denied** message will be displayed and the main ThinApp application will fail to load.

The following are limitations of the AppLink feature:

- ThinApp supports importing up to 250 packages at a time, and each package may be any arbitrary size.
- Packages that have been updated via AppSync will not have updates visible to the parent executable.
- Sandbox changes from packages being imported will not be visible to the parent executable.

Add AppLink Reference Dialog Box



Note • The AppLink Settings feature requires ThinApp 4.x. If you are using Thinstall 3.x, any AppLink settings that you define will be ignored.

The AppLink (Application Link) feature enables you to configure relationships between ThinApp applications that work together. On the **Add AppLink Reference** dialog box, which is opened by clicking the Browse button on the **AppLink Settings** dialog box, you specify the name and location of a ThinApp application and indicate whether that application is Required or Optional.

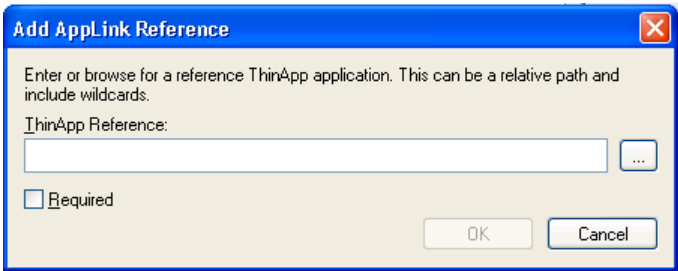


Figure 10-47: Add AppLink Reference Dialog Box

In the **ThinApp Reference** box, enter the relative (runtime) path to the existing ThinApp application that you want to link to. For more information on how to specify a ThinApp Reference, see the following:

- [Enter a Relative Path](#)
- [Path Name Format](#)
- [Which ThinApp File Should Be Specified in an AppLink Reference?](#)
- [Required vs. Optional](#)
- [Examples of AppLink References](#)

Enter a Relative Path

On the **Add AppLink Reference** dialog box, if you click Browse and browse for a ThinApp application, the absolute path to that application is entered, such as C:\Program Files\AppName\filename.exe. In that case, the main ThinApp application needs that linked application to be found at the specified absolute path location at runtime, which is unlikely. Therefore, it is recommended that you enter a relative path name.

Path Name Format

AppLink supports both URL and UNC path names.

Which ThinApp File Should Be Specified in an AppLink Reference?

If a ThinApp application has only one shortcut, it consists of a single executable. Therefore, you would obviously specify that executable file when creating an AppLink Reference.

However, when a ThinApp application has more than one shortcut, the ThinApp file that you specify in an AppLink Reference depends upon what tool you used to build the ThinApp application:

Table 10-40 • File to Specify in an AppLink Reference

Tool Used to Build ThinApp Application	# of Shortcuts	ThinApp Application File to Specify
AdminStudio or ThinApp	Only one	Specify the executable file (.EXE).

Table 10-40 • File to Specify in an AppLink Reference

Tool Used to Build ThinApp Application	# of Shortcuts	ThinApp Application File to Specify
AdminStudio	More than one	When built with AdminStudio, a ThinApp application that has more than one shortcut consists of two or more executable files and a Package.DAT file (as described in Components of a ThinApp Application). In this situation, specify the Package.DAT file.
ThinApp	More than one	When built with ThinApp, a ThinApp application that has more than one shortcut consists of multiple executable files, with one primary executable. In this situation, specify the primary executable file (.EXE).

Required vs. Optional

If you want this package to be required, select the **Required** option. If a required package is missing from the virtual package, it will fail to run. Note the following about required packages:

- If any specified package fails to import, an error message will be displayed and the parent executable file will exit.
- If a wildcard pattern is used to specify a package, no error message is displayed if no files match the wildcard pattern. Therefore, if a wildcard pattern is used to specify a package, the reference is always optional.
- To continue even if load errors occur, make the package references optional instead.


Examples of AppLink References

The following are examples of how packages can be added to the **AppLink References** list:

Table 10-41 • AppLink References Examples

Example	Description
Plugin.exe	This will import a single package located in the same directory as the parent executable.
plugins\Plugin.exe	This will import a single package located in the plugins subdirectory of the parent executable.

Table 10-41 • AppLink References Examples

Example	Description
plugins*.exe	<p>This will import all executables located in the plugins directory.</p>  <p>Important • If any executable fails to import because it is not a proper ThinApp package or because of a security issue, the parent executable will fail to load.</p>
n:\plugins*.exe	This will import all EXEs located at the absolute path n:\plugins.
%PLUGINS%*.exe	This expands the environment variable, PLUGINS, and imports all executables found at this location.
plugin1.exe;plugin2.exe;plugins*.exe	This loads two specified plugins and a list of executables found in the plugins subdirectory.

AppSync Settings Dialog Box



Note • The AppSync Settings feature requires ThinApp 4.x. If you are using Thinstall 3.x, any AppSync settings that you define will be ignored.

AppSync (Application Sync) enables you to automatically keep deployed virtual applications up to date. When an application starts up, AppSync can query a Web server to see if an updated version of the package is available. If an update is available, the differences between the existing package and the new package will be downloaded and used to construct an updated version of the package. The updated package will be used for future deployments.

You can use the AppSync feature to perform the following tasks:

- **Distribute runtime components separately**—You can use AppSync to distribute runtime components separately from the applications that use them. For example, the Java Runtime Environment (JRE) or ODBC drivers.
- **Apply layered service packs to applications**—You can use AppSync to apply layered service packs to your applications. Application Sync enables you to distribute service packs and roll back to previous versions, if necessary.

On the **AppSync Settings** dialog box, which is opened by clicking **AppSync Settings** on the **More Options** menu of the **Build Options** page, you can configure AppSync settings for your ThinApp application.

On the **Expiration** tab, you can specify that the ThinApp application is required to check for updates at a defined frequency. If the ThinApp application fails to successfully check for updates within that defined frequency, it will fail to run. Note that the update does not expire, the ThinApp application expires, and cannot be used until it is updated. successfully.

The **AppSync Settings** dialog box includes two tabs:

- General Tab
- Expiration Tab

General Tab

On the **General** tab, you specify the location of the Web server that hosts application updates.

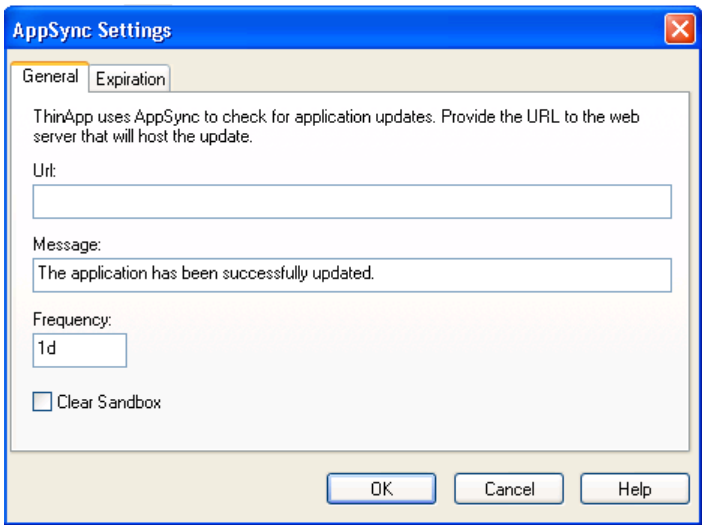


Figure 10-48: General Tab of the AppSync Settings Dialog Box

The following options are included:

Table 10-42 • General Tab of the AppSync Settings Dialog Box

Option	Description
Url	URL of the Web server where updates are stored. Application Sync works over both the HTTP (unsecure) and HTTPS (secure) protocol. Part of HTTPS is that the identity of the Web server is checked. You can include a user name and password in the URL that will be used for basic authentication. The standard Windows/Internet Explorer proxy setting is respected. For example: <code>https://example.com/some/path/PackageName.exe</code>
Message	When an updated package is first launched, an information message can be shown. For example: <code>Your application has been updated.</code>
Frequency	By default, a package will connect to the Web server once per day to see if an updated version is available. You can set the frequency by modifying this setting. For example, to set the Frequency to 2 days, enter <code>2d</code> . For 2 weeks, enter <code>2w</code> , etc.

Table 10-42 • General Tab of the AppSync Settings Dialog Box

Option	Description
Clear Sandbox	Gives you the option to clear the sandbox after an update. By default, the sandbox is not cleared. Select this option to clear the sandbox.

Expiration Tab

On the **Expiration** tab, you can specify that a ThinApp application is required to check for updates at a defined frequency. If the ThinApp application fails to successfully check for updates within that defined frequency, it will fail to run.

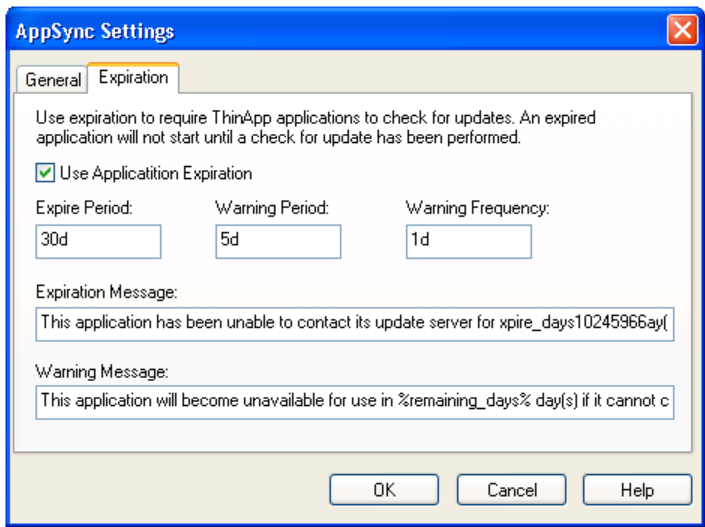


Figure 10-49: Expiration Tab of the AppSync Settings Dialog Box

The following options are included:

Table 10-43 • Expiration Tab of the AppSync Settings Dialog Box

Option	Description
Use Application Expiration	Select this option to require that an application has to check for updates at a specified frequency.
Expire Period	Sets the update frequency in minutes (m), hours (h), or days (d). If the Web server cannot be reached, the package will continue to work until the Expire Period is reached. This default setting is 30 days but you can change that setting by modifying this setting. For example: <ul style="list-style-type: none">• To set the period to 30 days, enter 30d• If you do not want the package to expire, clear the Use Application Expiration check box.

Table 10-43 • Expiration Tab of the AppSync Settings Dialog Box

Option	Description
Warning Period	Sets the start of the warning period before a package expires. For example, to set the period at 5 days, enter 5d .
Warning Frequency	<p>Sets the frequency of warnings before the package expires. With the default of one day, the warning message will be displayed once per day only. To configure the warning to pop up on every application launch, enter 0. To configure it to pop up every 4 days, enter 4d.</p> <p>After the warning period has started, the Web server will be checked on every launch of an application, overriding any previous setting.</p> <p>As long as a package has not expired, this parameter checks for new versions and downloads will occur in the background. The user can continue to use the old version. If the application is terminated by the user before the download is complete, the download will resume when a virtual application is launched again. After the download completes, the new version will be activated on the next launch.</p> <p>When the package has expired, the version check and download will happen in the foreground. A progress bar will be shown during the download phase.</p>
Expiration Message	<p>After the expiration limit has been reached and a virtual application is started, it will try to connect to the Web server and check for a new version. If the connection fails, a message box will be shown and execution will be terminated. The default message is shown in the example below.</p> <p>For example:</p> <p>This application has been unable to contact its update server for <i>Expire_Period</i> days, so it is unavailable for use. Check your network connection and try again.</p>
Warning Message	<p>If the connection to the Web server fails, a message box will be shown. The default message is:</p> <p>This application will become unavailable for use in <i>Warning_Period</i> days if it cannot contact its update server. Check your network connection to ensure uninterrupted service</p>



Note • If you use AppSync, VMware recommends that you disable automatic application updates that are configured in your virtual application. Conflicts might occur between the linked packages and the software that is automatically updated. If an automatic update feature updates an application, it stores the updates in the sandbox. If AppSync then updates the application to a different version, the updates stored in the sandbox take precedence over the files contained in the version that AppSync created. The order of precedence for the update files are those in the sandbox, then the virtual operating system, and then the physical machine.

Building ThinApp Applications Using the Command Line

When you configure a ThinApp application in an InstallShield project and then build that project (using either the user interface or the command line), both the Windows Installer package and the ThinApp application are built. When you use the standard InstallShield command line build, you do not need to add any additional command line parameters. All of the ThinApp application settings are saved within the InstallShield project.

ThinApp Application Conversion Error and Warning Messages

For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see [Virtualization Conversion Errors and Warnings](#).

Application Features Requiring Pre- or Post-Conversion Actions

Some application features are ignored when creating a ThinApp application. Therefore, some additional pre- or post-conversion actions must be taken in order for the ThinApp application to be created properly.

One action you could take to try to include ignored features in an ThinApp application is to first repackage the application using the Repackaging Wizard, and then convert the repackaged application to a ThinApp application.

For a list of ignored features, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

ThinApp Not Found

To create a ThinApp application, you are required to have both AdminStudio and ThinApp installed on the same machine. If a user attempts to create a ThinApp application without this ThinApp component, a message is displayed and the build is unsuccessful.

To purchase the ThinApp, visit the VMware Web site:

http://thinstall.com/products/virtualization_suite.php

ThinApp Application Configuration File: package.ini

ThinApp application configuration options that you set in the ThinApp Assistant interface are recorded in the `package.ini` file that is generated when the ThinApp application is built.

A `package.ini` contains the following groups of options:

- [\[BuildOptions\]](#)
- [\[Compression\]](#)
- [\[Isolation\]](#)
- [\[MainApp.exe\]](#)
- [\[Test.exe\]](#)



Note • For the latest information on the ThinApp application configuration file, *package.ini*, consult your ThinApp documentation.

[BuildOptions]

The [BuildOptions] section of the *package.ini* file specifies Global options which will be inherited by each child executable file. The following options are included:

Table 10-44 • [BuildOptions] Section of *package.ini*

Option	Description
SandboxName	<p>When a ThinApp application is built, a Sandbox cache is created in the following location:</p> <p>c:\Documents & Settings\USER_NAME\Application Data\ThinApp\SandboxName</p> <p>The SandboxName entry in the <i>package.ini</i> file is used to name the directory where sandbox files are stored at runtime.</p> <p>SandboxName=MyAppIicationV3</p> <p>By default, AdminStudio names the Sandbox by assigning it a unique GUID. However, if you want to override this default Sandbox name, you may (optionally) enter a new name in the <i>package.ini</i> file using the SandboxName option.</p> <p>If no Sandbox Name is entered, a unique GUID is used, such as:</p> <p>SandboxName={2BDBE10A-9E53-4B5E-811D-DF8019D0B13C}</p> <div></div> <p>Note • This option corresponds to the Sandbox Name field on the General Information page.</p>
InventoryName	<p>Used by desktop management systems to identify packages for usage reporting purposes. If you do not use a desktop management system or license-controlled system, this value has no effect</p> <p>InventoryName=MainApp v1.0</p>

Table 10-44 • [BuildOptions] Section of package.ini (cont.)




Option	Description
SandboxNetworkDrives	<p>Enable this option if you want changes to data on Network-mapped drives to go into the sandbox. By default, the ThinApp application can read and write to network mapped drives with no changes. The value for SandboxNetworkDrives is set to either 0 (off) or 1 (on).</p> <p>SandboxNetworkDrives=0</p>  <p>Note • This option corresponds to the Mapped Network Drive Changes go to Sandbox option on the General Settings page.</p>
SandboxRemovableDisk	<p>Enable this option if you want changes to data on Removable disk (floppy/flash) to go into the sandbox. By default the application can read and write to removable disk with no changes. The value for SandboxRemovableDisk can be set to either 0 (off) or 1 (on).</p> <p>SandboxRemovableDisk=0</p>  <p>Note • This option corresponds to the Removable Disk Changes go to Sandbox option on the General Settings page.</p>
RemoveSandboxOnExit	<p>Enable this option if you want to delete the sandbox when the ThinApp application exits. This resets the application to its original captured state. If the application spawns child processes, the clean up will be postponed until all have quit. The value for RemoveSandboxOnExit can be set to either 0 (off) or 1 (on).</p> <p>RemoveSandboxOnExit=0</p>  <p>Note • This option corresponds to the Reset Sandbox on Exit option on the General Settings page.</p>

Table 10-44 • [BuildOptions] Section of package.ini (cont.)


Option	Description
ExternalCOMObjects	<p>This option allows you to specify that you want specific COM objects to be executed on the system instead of in the virtual environment. This option only applies to out-of-process COM objects (LocalServer32) and Services-based COM objects.</p> <p>To specify multiple objects, put a semicolon after each entry. Objects should always be specified in CLSID format</p> <p>The following class ID specifies the class ID for Microsoft Word:</p> <pre>ExternalCOMObjects={000209FF-0000-0000-C000-000000000046};{000209FF-0000-0000-C000-000000000047}</pre>  <p>Caution • This option is for advanced users.</p>
VirtualizeExternalOutOfProcessCOM	<p>Enable this option if you want all out-of-process COM objects to be loaded outside of the virtual environment . By doing this, the application may indirectly modify the machine—for example, the MSI installer service COM object could be modified.</p> <pre>VirtualizeExternalOutOfProcessCOM=0</pre> <p>The value for this option can be set to either:</p> <ul style="list-style-type: none"> • 0—Inside virtual environment • 1—Outside the virtual environment <p>The default is to create all out-of-process COM objects inside the virtual environment.</p>
PermittedGroups	<p>Using this option, you can specify the Active Directory groups which are allowed to use this ThinApp application.</p> <pre>PermittedGroups=Group1;Group2;Group3</pre>  <p>Note • This option corresponds to the Allow application execution to the following user groups option on the General Settings page.</p>

Table 10-44 • [BuildOptions] Section of package.ini (cont.)


Option	Description
AccessDeniedMsg	<p>Use this option to customize the message the user sees if they do not have permission to execute a ThinApp application.</p> <p>AccessDeniedMsg=You do not have access to execute this application, please contact your Administrator</p>  <p>Note • This option corresponds to the Message shown when users not belonging to above groups run the ThinApp application field on the General Settings page.</p>
ChildProcessEnvironmentDefault and ChildProcessEnvironmentExceptions	<p>Executables located in the virtual file system are always executed within the virtual environment. Executables located in the physical file system can be executed inside or outside the virtual environment.</p> <p>The default is determined by the ChildProcessEnvironmentDefault option, which can be set to Virtual or External. If this option is not present, the default is the Virtual environment.</p> <p>It is possible to override the default for specific applications by specifying a list of applications, separated by semicolons, using the ChildProcessEnvironmentExceptions option. If a complete path is specified, the full name of the executable is used for the comparison; otherwise, only the file name is used.</p> <p>For example:</p> <pre>ChildProcessEnvironmentDefault=Virtual ChildProcessEnvironmentExceptions=exec.exe;c:\path\file.exe</pre> <p>In this example, c:\exec.exe, c:\Windows\exec.exe and c:\path\file.exe would be executed externally.</p>
AutoShutdownServices	<p>Use this option to specify if virtualized services keep on running when the last non-service process exits. Permitted values are:</p> <ul style="list-style-type: none"> • 0—Keep on running. • 1—Stop virtualized services (Default). <p>AutoShutdownServices=1</p>


Table 10-44 • [BuildOptions] Section of package.ini (cont.)

Option	Description
NetRelaunch	<p>Under some conditions, Norton AntiVirus will try to perform a complete scan of an executable. This scan can have a big impact on launch times for large executable files located on network shares. Norton AntiVirus decides to perform a complete scan under these conditions:</p> <ul style="list-style-type: none"> • If the executable is launched from a network share or removable disk. It skips the scan when the executable is located on the hard drive). • When the executable makes its first network connection. It does not scan the executable if the executable does not make any network connections. <p>Because a large number of desktops have Norton AntiVirus installed, ThinApp automatically compensates for this by allowing applications to launch from a network share without incurring the lengthy scan times. It does so by creating a small stub executable in the user's sandbox which is then relaunched. Because the small executable can be scanned quickly, it will load the remainder of the application data from the original source location.</p> <p>You can disable ThinApp default behavior by adding the NetRelaunch=1 option to disable full file scans.</p> <p>NetRelaunch=1</p>

[Compression]

The [Compression] options specify the default compression options to use when building the ThinApp application.

Table 10-45 • [Compression] Section of package.ini

Option	Description
CompressionType	<p>To reduce the application startup time, you can specify the CompressionType option to compress the ThinApp application.</p> <p>CompressionType=Fast</p> <p>Specify one of the following options:</p> <ul style="list-style-type: none"> • None: Do not perform any type of compression • Fast: Perform quick compression resulting in a smaller application footprint • Small: Perform the best compression resulting in the smallest application footprint, but increasing build time.  <p>Note • This option corresponds to the Compression Type options on the Build Options page.</p>

[Isolation]

The [Isolation] options specify the isolation options to use for folders and registry keys when building the ThinApp application.

Table 10-46 • [Isolation] Section of package.ini

Option	Description
DirectoryIsolationMode	<p>This option specifies the default isolation options to use for folders when building this project.</p> <p>DirectoryIsolationMode=WriteCopy Merged</p> <p>This option has the following possible values:</p> <ul style="list-style-type: none"> • WriteCopy—System elements are visible, modifications to both virtual and system elements are made in the sandbox, new elements are created in the sandbox, and if a system element and a virtual element are at the same location, the application sees the virtual element. • Merged—System elements are visible, modifications to virtual elements are made in the sandbox, modifications to system elements are made on the system, new elements are created on the system, and if a system element and a virtual element are at the same location, the application sees the virtual element.

Table 10-46 • [Isolation] Section of package.ini (cont.)

Option	Description
RegistryIsolationMode	<p>This option specifies the default isolation options to use for registry keys when building this project.</p> <p>RegistryIsolationMode=WriteCopy Merged</p> <p>This option has the following possible values:</p> <ul style="list-style-type: none"> • WriteCopy—System elements are visible, modifications to both virtual and system elements are made in the sandbox, new elements are created in the sandbox, and if a system element and a virtual element are at the same location, the application sees the virtual element. • Merged—System elements are visible, modifications to virtual elements are made in the sandbox, modifications to system elements are made on the system, new elements are created on the system, and if a system element and a virtual element are at the same location, the application sees the virtual element.

[MainApp.exe]

The [MainApp.exe] section specifies the source executable, the name of the file that contains read-only registry data to be bound, whether to perform logging, and the icon to use for the executable.

Table 10-47 • [MainApp.exe] Section of package.ini

Option	Description
Source	<p>This option specifies the .exe which will be run to launch the ThinApp application.</p> <p>Source=%ProgramFiles%\Test\MainApp.exe</p> <p>This option also specifies the icon that will be used, if an icon is not explicitly specified using the Icon option.</p>
ReadOnlyData	<p>This option specifies the name of the file that contains read-only registry data to be bound. If the read-only registry also has an associated file-data, the file-data file should be in the same directory with the appended extension TestMain.exe.ro.thfd.</p> <p>ReadOnlyData=bin\MainApp.exe.ro.tvr</p>

Table 10-47 • [MainApp.exe] Section of package.ini (cont.)


Option	Description
DisableTracing	<p>This optional setting will disable logging/tracing capabilities for this application when Log Monitor is running. Possible values are 1 (logging is disabled) or 0 (logging is enabled).</p> <p>DisableTracing=1</p>  <p>Note • This option corresponds to the Disable Log Monitor Tracing option on the Build Options page.</p>
Icon	<p>By default the icon is used from the executable identified in the Source option. You can change this to specify one of the following:</p> <p>Icon=SomeOtherEXE.exe</p> <p>Icon=NULL</p> <p>Icon=SomeOtherIco.ico</p>

Table 10-47 • [MainApp.exe] Section of package.ini (cont.)

Option	Description
RetainAllIcons	<p>By default, each application retains the main Group Icon from its Source executable and the individual icon resource pointed to by the Group Icon. Tlink will strip out extra icons that cannot be used directly by the system shell. However, you can force these extra icons to be included in the ThinApp executable by using the RetainAllIcons=1 option. For example:</p> <pre>[myapp.exe] Source=%ProgramFilesDir%\myapp\app.exe RetainAllIcons=1</pre> <p>Instead of using the Source option to identify your application icon, you can also use:</p> <ol style="list-style-type: none"> 1. The value NULL. In this case, the application will not have an icon and Windows will use the default application icon. <pre>[myapp.exe] Source=%ProgramFilesDir%\myapp\app.exe Icon=NULL</pre> 2. The path to another .exe file. In this case, Tlink will load the icons from a different .exe file. If a full path is not specified, the path is relative to the project directory. <pre>[myapp.exe] Source=%ProgramFilesDir%\myapp\app.exe Icon=%ProgramFilesDir%\myapp\app2.exe</pre> <p>Executable files can contain multiple icon sets. You can optionally specify which set to use by appending ",1" ",2" to the end of the Icon path name like this:</p> <pre>[myapp.exe] Source=%ProgramFilesDir%\myapp\app.exe Icon=%ProgramFilesDir%\myapp\app2.exe,1</pre> 3. The path to an .ico icon file. In this case, Tlink will load the icons from the specified .ico file. If a full path is not specified, the path is relative to the project directory. <pre>[myapp.exe] Source=%ProgramFilesDir%\myapp\app.exe Icon=%ProgramFilesDir%\myapp\myicon.ico</pre>

[Test.exe]

The [MainApp.exe] section specifies the Shortcut and WorkingDirectory options.

Table 10-48 • [Test.exe] Section of package.ini

Option	Description
Shortcut	<p>The Shortcut option specifies whether the .exe that is generated will contain any registry or file data. This information will be loaded from the .exe referenced by the Shortcut option.</p> <p>Shortcut applications can specify WorkingDirectory and CommandLine</p> <p>Shortcut=MainApp.exe</p>
WorkingDirectory	<p>The WorkingDirectory option specifies where the ThinApp application will start. If this option is not specified, the Current Working Directory will be inherited from the parent process</p> <p>WorkingDirectory=%ProgramFiles%\Test</p>

Creating Citrix Profiles

You can use the Citrix Assistant to help you author a Citrix profile for an application. The Citrix profile can then be deployed on a Citrix XenApp. These deployed applications run within isolation environments that prevent them from interfering with other software running on the same machine. Using the Citrix Assistant, you can configure a Citrix profile's operating system and language requirements, files, folders, shortcuts, registry settings, script execution, isolation options, and build options.

Information about creating Citrix profiles using the InstallShield Citrix Assistant is organized into the following sections:

- [Overview of the Citrix Assistant](#)
- [Using the Citrix Assistant to Create a Citrix Profile](#)
- [Citrix Assistant Reference](#)

Overview of the Citrix Assistant


You can use the Citrix Assistant to help you author a Citrix profile for an application. The Citrix profile can then be deployed on a Citrix XenApp. These deployed applications run within isolation environments that prevent them from interfering with other software running on the same machine. Using the Citrix Assistant, you can configure an application's operating system and language requirements, files, folders, shortcuts, registry settings, script execution, isolation options, and build options.

The process for authoring a Citrix profile using the Citrix Assistant is as follows:

Table 10-49 • Steps to Convert a Windows Installer Package to a Citrix Profile

Step	Go To:	Actions
Getting Started	InstallShield Start Page	Create or open one of the following project types: <ul style="list-style-type: none"> • Basic MSI • MSI Database (Direct Edit Mode) • Transform (Direct MST Mode)
	Citrix Assistant Home Page	Click on the Citrix XenApp tab to open the Citrix Assistant Home page
Specifying Citrix Profile Information	Profile Information Page 	Specify the name and version of the Citrix profile, whether this package can run executables that are not included with the Citrix profile, and whether to include diagnostic tools with the Citrix profile.
Specifying Operating System and Language Requirements	Profile Requirements Page 	Specify the operating systems and language requirements that client workstations must meet in order for this application to operate properly. You can also specify pre-launch and post-exit scripts to execute.
Managing Files and Folders in a Citrix Profile	Profile Files Page 	View existing files and folders, add and delete files.
Setting Isolation Options	Profile Files Page 	Override the Citrix default isolation options for selected folders and files. Isolation options specify how the virtual environment will provide access to files and folders requested by the Citrix profile.
Modifying Profile Shortcut Settings	Profile Shortcuts Page 	Create, delete, include, exclude, or rename a Citrix profile's executables, which are derived from the shortcuts in its Windows Installer package.
Modifying Profile Registry Settings	Profile Registry Page 	Add, delete, or modify the registry settings in your Citrix profile, and override the Citrix default isolation options for selected registry keys. Isolation options specify how the virtual environment will provide access to registry keys requested by the Citrix profile.

Table 10-49 • Steps to Convert a Windows Installer Package to a Citrix Profile

Step	Go To:	Actions
Modifying Build Settings	Build Settings Page 	<p>Choose whether to digitally sign the Citrix profile and select the releases that you want to build.</p> <p>Also, when you have a Windows Installer package open in Direct Edit mode, you can enable the Build Release option on the Build menu by making a selection on this page.</p>
Building a Citrix Profile	Build on the Toolbar OR Build Citrix Profile (F7) on the Build Menu	<p>Click Build to build the active Release and create a Citrix profile.</p> <p>Also, when you have a Windows Installer package open in Direct Edit mode, you can enable the Build Release option on the Build menu by selecting the Build Citrix Profile option on this page.</p>

Information about the Citrix XenApp and Citrix profiles is presented in the following topics:

- [About Citrix XenApp](#)
- [About the Citrix Assistant](#)
- [About Citrix Profiles](#)
- [Benefits of Deploying Citrix Profiles](#)
- [Supported InstallShield Project Types](#)
- [How Transforms are Included in a Citrix Profile](#)



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

About Citrix XenApp

Citrix XenApp is an application delivery system for Windows applications that offers both application virtualization and application streaming. Applications are centralized on the Citrix XenApp, and then those applications are deployed to users throughout the enterprise. These deployed applications run within isolation environments that prevent them from interfering with other software running on the same machine.

Citrix XenApp: 2 Steps to Application Delivery

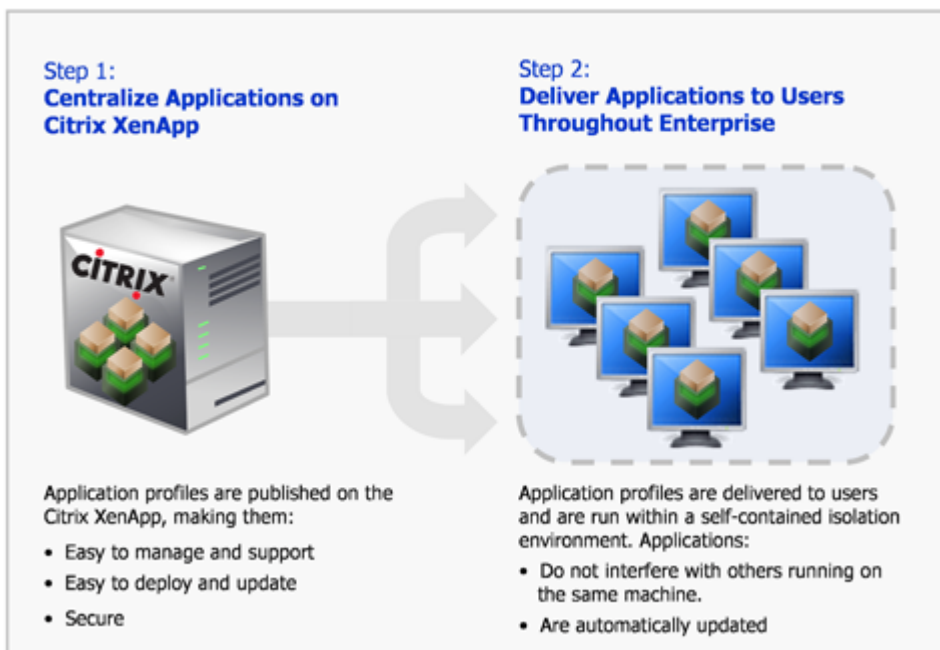


Figure 10-50: Citrix XenApp: Two Steps to Application Delivery

When applications are deployed on a Citrix XenApp, users can run those applications in an isolation environment, without installing, while connected or offline. Applications behave just like they were installed locally, but without any of the problems of installation, such as interfering with other applications on the same device. Files are saved locally and individual settings are preserved. Every time the application is run, it checks for errors or updates and they are delivered automatically.



Note • For more information, see [Benefits of Deploying Citrix Profiles](#).

About the Citrix Assistant

You can use the Citrix Assistant to prepare a Windows Installer package for deployment on Citrix XenApp by converting it to a Citrix profile. During this process, you:

- **Profile Information**—Specify profile information.
- **OS and Language Requirements**—Specify the operating system and language requirements for the application.
- **Files, Folders, Shortcuts, Registry Settings**—Specify files, folders, shortcuts, and registry settings included in application.
- **Isolation Options**—Define a set of options for running the application in isolation on the user desktop.
- **Build**—Specify build settings and build a Citrix profile.

The following diagram illustrates the Citrix profile creation process:

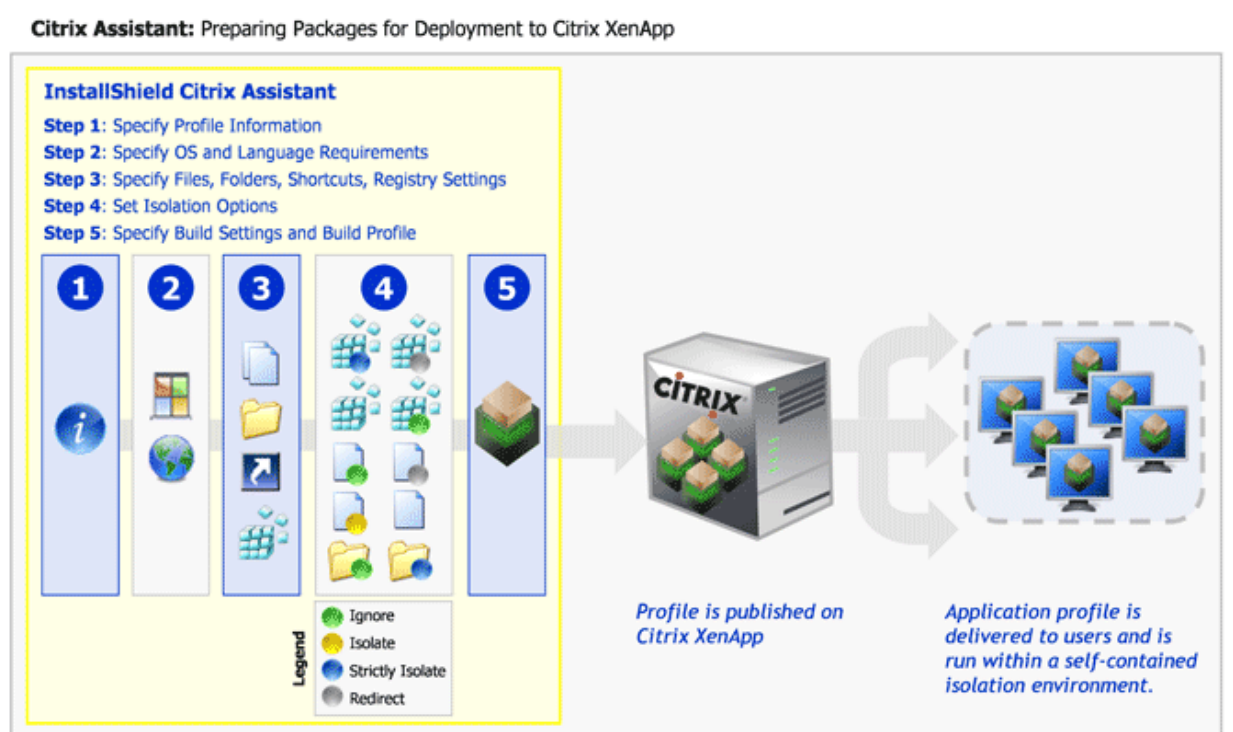


Figure 10-51: Preparing Packages for Deployment on Citrix XenApp



Note • You can also convert a Windows Installer package to a virtual application using Repackager. See [Converting Windows Installer Packages to Virtual Applications](#).

About Citrix Profiles

When you use the Citrix Assistant to prepare a Windows Installer package for deployment on the Citrix XenApp, the resources you generate are called *profiles*. A profile consists of the following files and directories:

Table 10-50 • Components of an Citrix Profile

Component	Name	Description
Profile Manifest File	myapp.profile	An XML file that defines the profile.
CAB File	[alphanumeric_string].cab	Compressed cabinet file that provides the isolation environment contents for the application.
Hashes File	Hashes.txt	Hash key file for digital signatures and signing profiles.

Table 10-50 • Components of an Citrix Profile

Component	Name	Description
Icons File	Icons.bin	Icons repository.
Scripts Folder	Scripts	Folder containing any pre- launch or post-exit scripts that you have chosen to include.



Caution • *Modifying these files directly is **not recommended**. To make any modifications, use the Citrix Assistant.*

These files are saved in a directory named `CitrixProfile`. The location of the `CitrixProfile` directory depends upon the type of file you are editing in InstallShield:

- **InstallShield project**—The `CitrixProfile` directory will be located in a subdirectory of the directory that contains this InstallShield project file, such as:

`C:\InstallShield 2008 Projects\ProductName\ConfigurationName\ReleaseName\CitrixProfile`

- **Windows Installer package**—The `CitrixProfile` directory will be located in the same directory as the Windows Installer file, such as:

`C:\FolderContainingMSI\CitrixProfile\ProductName`

The contents of the application profile are published on the Citrix XenApp.

A profile can contain a single application or suite of applications. For example, you can profile Microsoft Word by itself, or you can profile the entire Microsoft Office suite in a single profile.

Benefits of Deploying Citrix Profiles

Converting a Windows Installer package to a Citrix profile and deploying it on a Citrix XenApp offers the following benefits:

- Reduces Application Conflicts
- Enables Rapid, Low Cost Application Deployment
- Enables Automatic Software Updates
- Centralized Application Management Provides Controlled Access and Security
- Enables User-Based Application Access Rather Than Machine-Based Access

Reduces Application Conflicts

Traditionally to deploy an application throughout an enterprise, the application was installed on each user's desktop. Therefore, prior to installation, each application had to be tested for conflicts against each target desktop image (operating system with existing applications). After resolving conflicts that were found during testing, each application then had to be installed on each desktop. This process was very time consuming not only during initial installation, but also when applying patches or upgrading.

Citrix profiles run within isolation environments, which separate the interaction between an application and the underlying operating system's resources in order to prevent the applications from interfering with others running on the same machine. Because applications do not interact, the need to perform any conflict analysis and regression testing prior to deployment is eliminated. This not only results in rapid application deployment, but it also reduces the total cost of application delivery, due to decreased labor by IT.

Also, because users running applications in an isolation environment encounter no conflicts with other applications, user calls to the help desk are decreased.

Enables Rapid, Low Cost Application Deployment

Deploying Citrix profiles on Citrix XenApp simplifies the deployment of new applications, updates and patch deployment, regardless of the diversity of the access devices, software languages, computing architectures, and networks that are involved.

- **Only a single instance of the application is installed**—Instead of deploying, managing, updating and securing a vast array of heterogeneous client software on each individual user's access device, a single instance of the application is installed on the Citrix XenApp. The IT department only has to test for one environment, and deploy and update in one place. This reduces the cost of application installation and support. Also, you can deploy a Citrix profile once on a Citrix XenApp and replicate it to other Citrix XenApps within the existing enterprise infrastructure.
- **Prevents application-specific server silos**—Deploying applications on Citrix XenApp prevents the build-up of application-specific server silos because you can safely install and reliably run multiple application versions and incompatible applications on the same server.
- **Enables you to quickly install and update software throughout your enterprise**—Because you can manage the delivery of all of your Windows-based applications from one centralized location, there is no need to go from desktop to desktop, travel from office to office, or wait for laptops to return to headquarters in order to install or update software. With Citrix XenApp, you can deliver applications and updates instantly anywhere, any time—to offshore employees, outsourcers, new branch offices, new mergers and acquisitions, and mobile workforces.

Enables Automatic Software Updates

When an upgrade or patch needs to be deployed, you would only need to update the Citrix profile on the Citrix XenApp, which will then automatically update all of the instances of that Citrix profile throughout the enterprise. This means that users always have the latest application updates and patches, automatically.

Centralized Application Management Provides Controlled Access and Security

With Citrix XenApp, you can centralize applications and data in secure data centers, which increases data security and ensures fast, reliable performance. Centralized application management using Citrix XenApp provides the following benefits:

- **Enhances security**—Enables you to control, protect, and retain intellectual property centrally to reduce the chance for data loss and theft. Citrix XenApp helps you prevent data from leaving the data center without your explicit permission, which supports regulatory compliance and security objectives. You can provide authorized access to appropriate users—such as employees, customers, and partners—while verifying the ongoing security of the environment.

- **Can provide managed access to applications to users outside of your organization**—You can standardize the use of applications, without having to standardize the machines that the applications use. This enables you to provide managed access to applications from computers that are not your own corporate assets, such as from contractor or consultant computers.
- **Monitors application usage and performance**—Citrix XenApp gives you end-to-end visibility into application usage and performance. It gives IT administrators the power to understand who is using what, how often, and to what extent. They can observe, monitor, measure, audit, report and archive all the dimensions of information flow throughout the computing environment. This enables informed decisions regarding application consolidation and retirement, capacity planning, service level agreements and departmental charge-back
- **Enables identity-driven access**—Citrix XenApp enables you to provide identity-driven access tailored to any user environment. It automatically analyzes the user's permissions and then delivers the appropriate level of access to applications without compromising security. Depending on who and where users are and what device and network they're using, they may be granted different levels of access. You can also easily "decommission" applications by simply turning off a user's permission to it.

Enables User-Based Application Access Rather Than Machine-Based Access

Users can access their applications anywhere on the network, regardless of where they are or what device they are using.

Supported InstallShield Project Types

The **Citrix XenApp** tab is available when one of the following InstallShield project types is open:

- Basic MSI Project
- MSI Database (Direct Edit Mode)
- Transform (Direct MST Mode)

How Transforms are Included in a Citrix Profile

The Citrix Assistant supports the inclusion of transform files with Windows Installer packages in a Citrix profile.

- **How transforms are applied during profile generation**—When building a Citrix profile, transforms that you have specified are automatically applied to the base Windows Installer (.msi) package to create a temporary package, and then the Citrix profile is generated from that temporary package.
- **Creating a new transform**—You can create a new transform in InstallShield, and then build a profile from that transform file. When you create a new transform file in InstallShield, you specify the root .msi file in the **Open Transform** wizard. The steps you take to generate a profile after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.
- **Converting a Windows Installer package with existing transforms**—If you have a Windows Installer package and one or more existing transform files, and you want to include these transforms in the Citrix profile, you need to open one of the *transforms* in InstallShield (rather than the .msi file). The **Open Transform** wizard will open, and you will be prompted to specify the root .msi file and which of the existing

.mst files you want to include. The steps you take to generate a profile after using that wizard are exactly the same as if you were editing a Windows Installer package in Direct Edit mode.



Caution • All of the transforms that you add to a Citrix profile must be located in the same folder as the Windows Installer .msi package so that they can be accessed when the profile is built.

Using the Citrix Assistant to Create a Citrix Profile

The steps you need to take to create a Citrix profile are the following:

Table 10-51 • Steps to Take to Create a Citrix Profile Using the Citrix Assistant

Step #	Description
Step 1	Specifying Citrix Profile Information
Step 2	Specifying Operating System and Language Requirements
Step 3	Managing Files and Folders in a Citrix Profile
Step 4	Setting Isolation Options
Step 5	Modifying Profile Shortcut Settings
Step 6	Modifying Profile Registry Settings
Step 7	Modifying Build Settings
Step 8	Building a Citrix Profile

Specifying Citrix Profile Information

When creating a Citrix profile, you need to specify the **Name**, **Description**, and **Version** of the Citrix profile. You also need to specify whether this package can run executables that are not included with the Citrix profile, and whether to include diagnostic tools with the Citrix profile. The following tasks are performed on the **Profile Information** page of the **Citrix Assistant**:

- Specifying the Profile Name, Description, and Version
- Enabling a Citrix Profile to Run Non-Included Executables
- Including Diagnostic Tools With an App-V Application

Specifying the Profile Name, Description, and Version

On the **Profile Information** page of the Citrix Assistant, you name the Citrix profile, and provide a description and version number.



Task: *To specify the Citrix profile name, description, and version:*

1. In the **Citrix Assistant**, open the **Profile Information** page.
2. In the **Name** field, enter a name for this Citrix profile. The name you enter here determines the file name of the generated Citrix profile.



Tip • *Do not include the version number in the profile name.*

3. In the **Description** field, enter a brief explanation of the purpose of this package. This information is stored as package metadata.
4. In the **Version** field, enter the version number of this Citrix profile. This information is stored as package metadata.
5. On the **File** menu, click **Save** to save your changes.



Enabling a Citrix Profile to Run Non-Included Executables

On the **Profile Information** page, you can enable a Citrix profile to run executables that are not included with the Citrix profile.



Task: *To enable a Citrix profile to run non-included executables:*

1. In the **Citrix Assistant**, open the **Profile Information** page.
2. To specify whether this package can run executables that are not included with the Citrix profile, select one of the following options under **Security Settings**:

Option	Description
Enhanced	<p>Select this option if you do not want to permit any executable files to run other than those included in the Citrix profile. For example, if the client is running an Internet Explorer plug-in included in the Citrix profile, the enhanced security setting prevents the client from running any other plug-ins that the user might download.</p> <p>You would choose Enhanced security to disable an application from running executable files it writes to its current working directory on the client.</p> <div> Caution • Choosing Enhanced security might restrict or prevent the operation of some applications.</div>
Relaxed	<p>Select this option to allow non-included executables that are accessed through the Citrix profile to run.</p> <p>You would choose Relaxed security to allow an application to run executable files it writes to its current working directory on the client.</p> <div> Caution • Choosing Relaxed security can enable the client to download and run potentially malicious software.</div>

3. On the **File** menu, click **Save** to save your changes.

Including Diagnostic Tools With a Citrix Profile

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **Profile Information** page, you can choose to include the Registry Editor and the Windows Command Prompt diagnostic tools with your Citrix profile.

If you include diagnostic tools with your Citrix profile, you will be able to look at the registry or file system for the application while it is running in its isolation environment. For example, if you were running a Citrix profile and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.

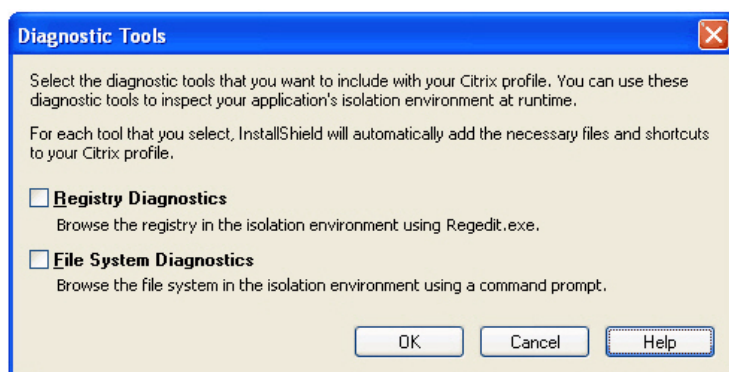


Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the Citrix profile. However, these tools may not be compatible with other operating systems.



Task: *To include diagnostic tools with a Citrix profile:*

1. In the **Citrix Assistant**, open the **Profile Information** page.
2. In the **More Options** list, click **Diagnostic Tools**. The **Diagnostic Tools** dialog box opens.



3. If you want to include the Registry Editor (`regedit.exe`) with your Citrix profile so that you can browse the profile registry at runtime from within the isolation environment, select the **Registry Diagnostics** option.
4. If you want to include the Windows Command Prompt application with your Citrix profile so that you can browse the virtual file system at runtime from within the isolation environment, select the **File System Diagnostics** option.

Launching the Diagnostic Tools Within the Isolation Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the profile.

When the user runs this Citrix profile application, two additional shortcuts will be available in the application's shortcut folder: The names of these shortcuts will reflect the application name, such as:

```
[ProductName] Registry  
[ProductName] File System
```

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application's Citrix isolation environment.

Specifying Operating System and Language Requirements

The next step in creating a Citrix profile is to open the **Profile Requirements** page and specify the operating system and language requirements that client workstations need in order to run the application locally.

Some applications can run on multiple operating systems and languages, while others, such as custom applications, might be able to run only on a particular operating system or language. When creating a profile, you need to customize it for the supported operating systems and languages .

Information about specifying operating system and language requirements includes the following topics:

- [Setting Operating System Requirements and Service Pack Levels](#)
- [Setting Language Requirements](#)
- [How Requirements are Applied at Runtime](#)
- [Adding Pre-Launch and Post-Exit Scripts](#)

Setting Operating System Requirements and Service Pack Levels

To specify the operating system and service pack level requirements for your application, perform the following steps.



Task: *To specify operating system requirement and service pack levels:*

1. In the **Citrix Assistant**, open the **Profile Requirements** page.
2. For the **Does your Citrix profile have any specific operating system requirements?** option, select one of the following:
 - **No**—Select this option if this application will run on all of the listed operating systems (which are the operating systems that the Citrix client supports). When this option is selected, the operating system check boxes are locked and cannot be changed.
 - **Yes**—Select this option if the application does not support one of the listed operating systems. When you select this option, the check boxes are unlocked and you can clear the selection of the unsupported operating systems.
3. If you set the previous option to **Yes**, do the following:
 - a. Select the operating systems that this application supports, and clear those that this application does not support.
 - b. For each of the selected operating systems, double-click on it and select **Service Packs Requirement** from the context menu to open the **Service Packs Requirements** dialog box, and choose one of the following options:
 - **No Service Pack Requirement**—This application supports all versions of this operating system, regardless of the number of Service Packs installed.

- **No Service Pack Allowed**—This application only supports the initial release of this operating system; if any Service Packs are installed, this application will not run properly.
- **Exact Service Pack Level**—This application requires the installation of a specific Service Pack on this operating system in order to run properly. Enter the required Service Pack Level in the box.
- **At Least Service Pack Level**—To run properly, this application requires that this operating system have at least the specified Service Pack (or higher) installed. Enter the minimum required Service Pack Level in the box.
- **At Most Service Pack Level**—To run properly, this application requires that this operating system have at most the specified Service Pack (or lower) installed. Enter the maximum required Service Pack Level in the box.
- **Range of Service Pack Levels**—To run properly, this application requires that this operating system have a specified range of Service Packs installed. If you select this option, specify the **Minimum Level** and **Maximum Level** in the boxes.

Setting Language Requirements

To specify language requirements for your application, perform the following steps.



Task: *To specify operating system requirement and service pack levels:*

1. In the **Citrix Assistant**, open the **Profile Requirements** page.
2. For the **Does your Citrix profile have any specific language requirements?** option, select one of the following:
 - **No**—Select this option if this application will run on all of the listed languages (which are the languages that the Citrix client supports). When this option is selected, the language check boxes are locked and cannot be changed.
 - **Yes**—Select this option if the application does not support one of the listed languages. When you select this option, the check boxes are unlocked and only **English** is selected by default.
3. If you selected **Yes** in the previous step, select only those languages that this application supports.

How Requirements are Applied at Runtime

The requirements you specify on the **Profile Requirements** page determine how, or if, a user has access to the application.

When a user attempts to run an application, the Citrix XenApp checks to see whether that user's workstation meets the profile's specified requirements. Then, depending upon the **Application Type** assigned to that profile when it was published on the server, the user is:

- granted access to run the application locally, *or*
- granted access to run the application from the server, *or*
- denied access to the application.

The user access scenarios are presented in the following table:

Table 10-52 • Citrix XenApp User Access Scenarios

Application Type	User Access to Application
Accessed from a server	User runs the application on the Citrix XenApp, using shared server resources.
Streamed if possible, otherwise from a server	<p>User access depends upon whether their workstation meets the profile's specified requirements:</p> <ul style="list-style-type: none"> • Meets requirements—The profile is streamed (copied) to the user's workstation, and the user runs the application locally (from within its isolation environment). • Does not meet requirements—User runs the application on the Citrix XenApp, using shared server resources.
Streamed to client	<p>User access depends upon whether their workstation meets the profile's specified requirements:</p> <ul style="list-style-type: none"> • Meets requirements—The profile is streamed (copied) to the user's workstation, and the user runs the application locally (from within its isolation environment). • Does not meet requirements—User cannot access the application.



Caution • If an application has specific operating system or language requirements and you fail to specify them correctly when creating the profile, users who do not meet those requirements will be given access to run applications locally and they will probably encounter application errors.

Adding Pre-Launch and Post-Exit Scripts

You can choose to include scripts with your profile that must execute either before profile launch or after profile exit in order for your application to run properly. On the **Script Execution** dialog box, which is opened by clicking **Script Execution** in the **More Options** list on the **Profile Requirements** page, you can view and manage all of the **Before Profile Launch** and **After Profile Exit** script files you are including with your Citrix profile.

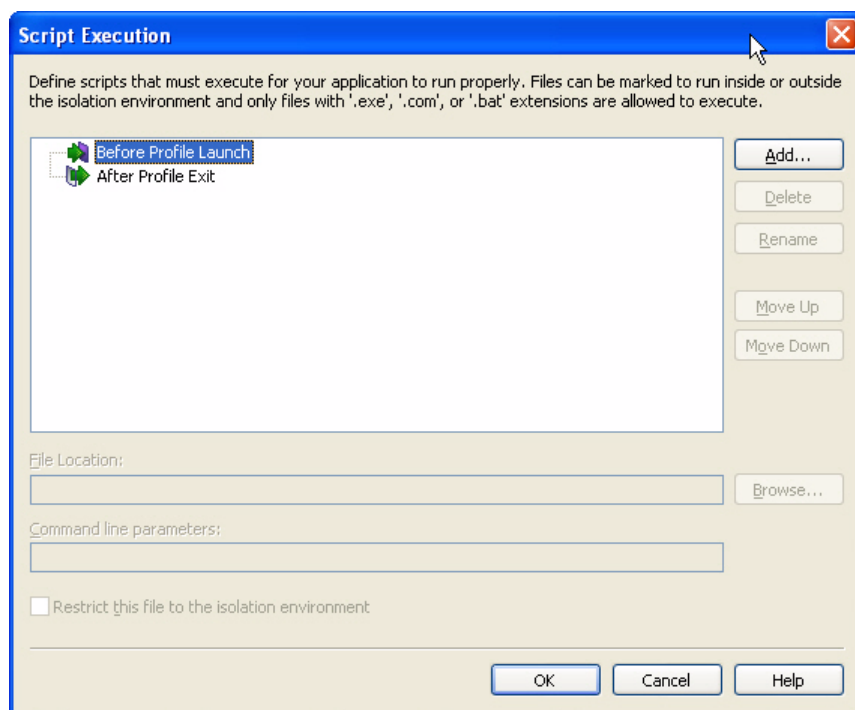
- Files can be marked to run inside or outside of the isolation environment.
- Only files with .exe, .cmd, .com, or .bat extensions are allowed to execute.

To add a script to your Citrix profile, perform the following steps.

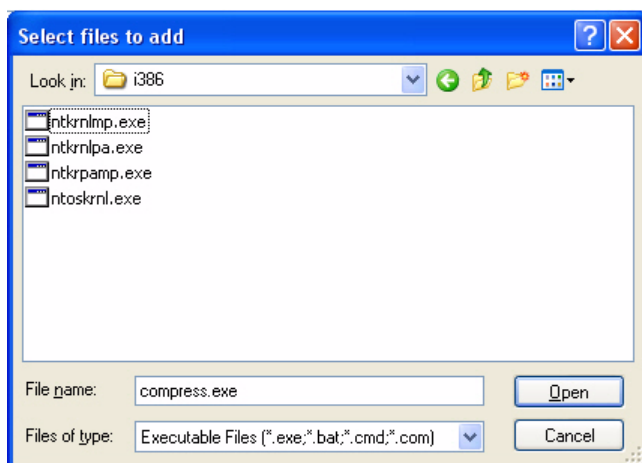


Task: *To add a before launch or after exit script to your Citrix profile:*

1. Open the **Profile Requirements** page of the Citrix Assistant.
2. In the **More Options** list, click **Script Execution**. The **Script Execution** dialog box opens.



3. Select the **Before Profile Launch** or **After Profile Exit** node in the tree.
4. Click **Add...**. The **Select Files to Add** dialog box opens.

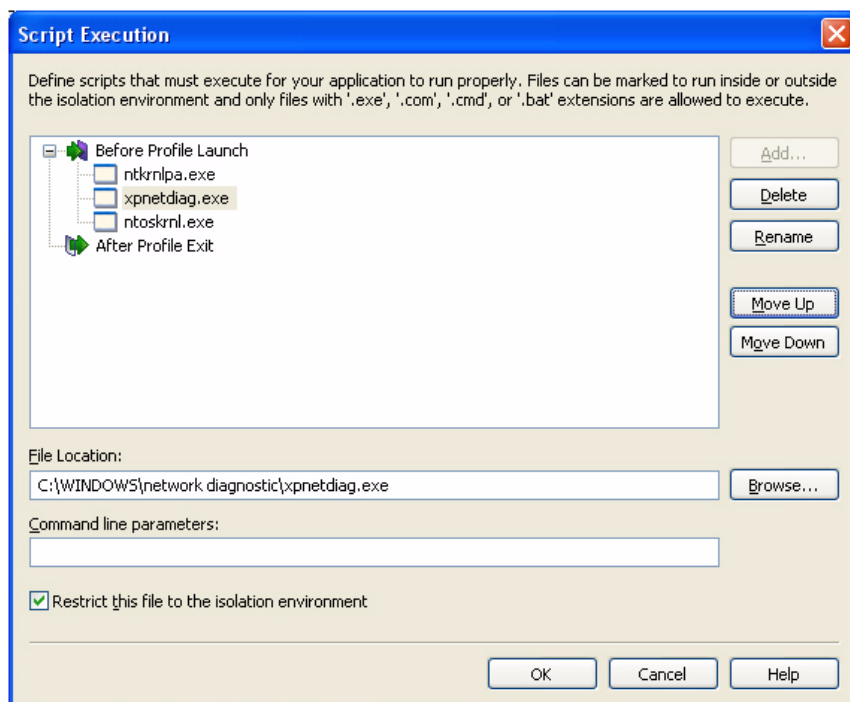


5. Select the script file(s) (.exe, .bat, .cmd, or .com) that you want to add, and click **Open**. The file is added to the Script Execution tree on under the appropriate node.



Tip • Use the Shift key to select multiple contiguous files, and use the Ctrl key to select multiple non-contiguous files.

6. Select a script in the tree. Several new fields and options are enabled.



You can now perform any of the following tasks:

- **Rename the file's display name**—To rename the script file's display name, click the **Rename** button and enter a new name. The name that is displayed on this dialog box to identify the script is changed, but the original name of the script file is not changed.
- **Select a different script**—To select a different script, click the Browse button and select a different script file (.exe, .bat, .cmd, or .com).
- **Reorder scripts**—If multiple scripts are listed under a node, you can use the **Move Up** and **Move Down** buttons to change the order that the scripts will be run. You can also reorder the scripts using the Ctrl+Shift+Up Arrow and Ctrl+Shift+Down Arrow keys.
- **Restrict script to isolation environment**—If you want this script to only be able to run within the Citrix profile's isolation environment, select the **Restrict this file to the isolation environment** option.
- **Add command line parameters**—To add command line parameters to run along with the script, enter them in the **Command line parameters** box.
- **Delete a script**—To delete a script from the profile, select it and click the **Delete** button.

7. When you have set all desired options for the script, click **OK**.

Managing Files and Folders in a Citrix Profile

The next step in creating a Citrix profile is to view existing files and folders, add and delete files and folders, and override the default isolation options for folders and files.

The following tasks are performed on the **Profile Files** page.

- [Managing Files and Folders in a Citrix Profile](#)
- [Controlling the Display of Predefined Folders](#)
- [Overview of Citrix Isolation Options](#)

Managing Files and Folders in a Citrix Profile

The directories in the destination tree on the **Profile Files** page of the Citrix Assistant represent how your application will be organized within its isolation environment.

On the **Profile Files** page, you can view all of the files and folders that are currently in your Citrix profile, add new files and folders to include in the Citrix profile, and delete files and folders from the Citrix profile.

- [Adding Files to a Citrix Profile](#)
- [Adding an Existing Folder \(and its Contents\) to a Citrix Profile](#)
- [Creating a New Folder](#)
- [Moving Files and Folders](#)
- [Deleting Files and Folders](#)

Adding Files to a Citrix Profile

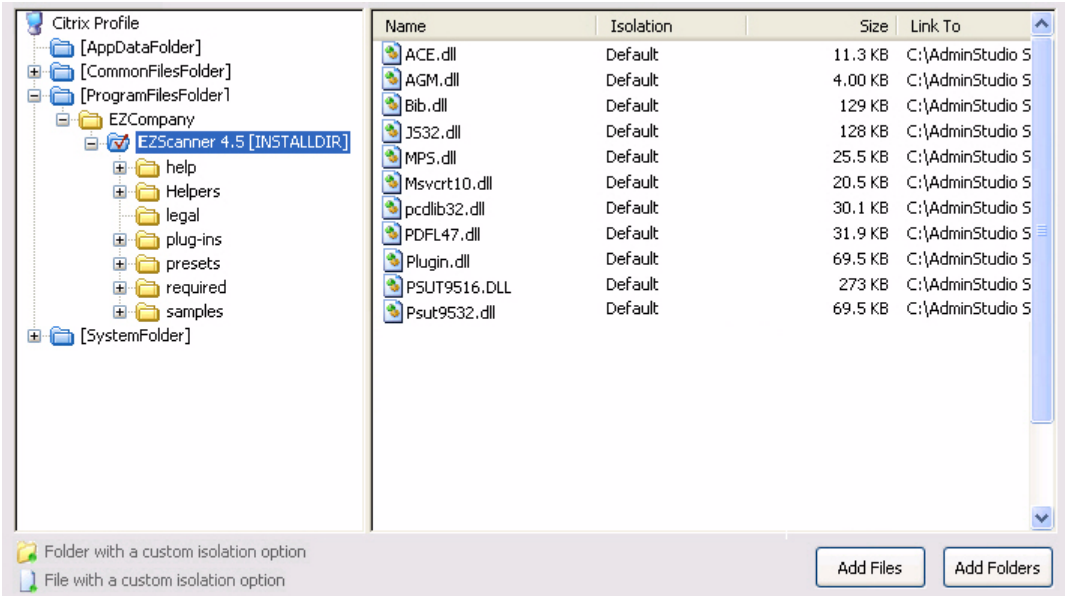
To add files to a Citrix profile, perform the following steps:



Task:

To add a files to a Citrix profile:

1. In the **Citrix Assistant**, open the **Profile Files** page. The files and folders are listed in the **Citrix Profile** tree, organized by installation directory.



- Folders are listed in the column on the left, and all of the files in the selected folder are listed on the right. Blue folders are the supported MSI standard folders. The folder with the check mark is INSTALLDIR, which represents the main product installation directory.
2. Browse through the folder tree to find the folder that you would like to add files to.
 3. Select the folder and click the **Add Files** button. The **Open** dialog box opens.
 4. Select the file or files that you want to add and click **Open**. The files you selected are now listed.



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Adding a File by Dragging and Dropping Files From Your System

You can also add files or folders to your Citrix profile on the **Profile Files** page by dragging them from a directory on your computer to the desired location in the tree.

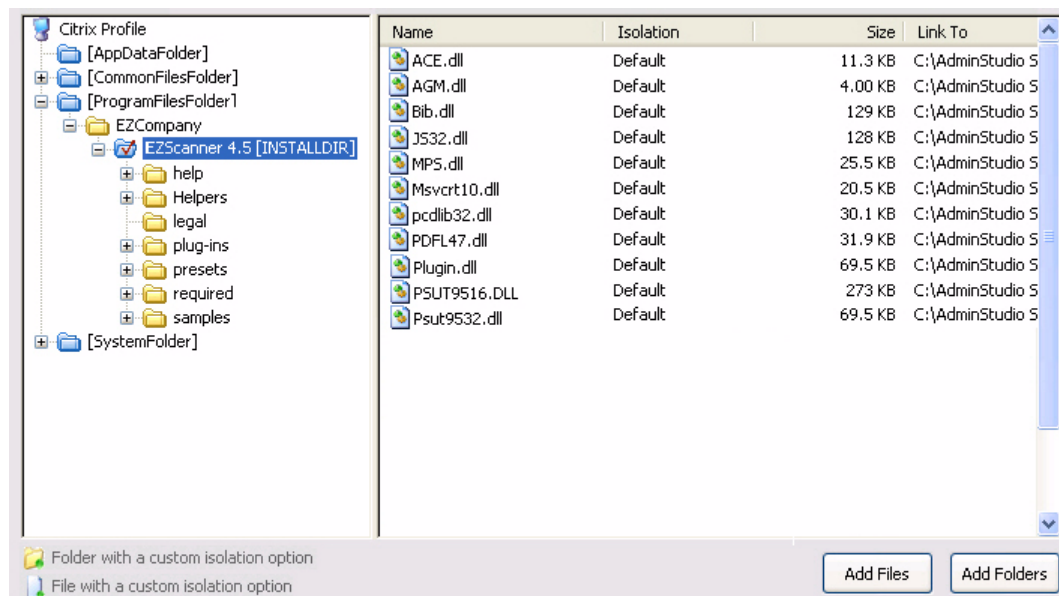
Adding an Existing Folder (and its Contents) to a Citrix Profile

To add an existing folder and all of the files and subfolders within it to a Citrix profile, perform the following steps:

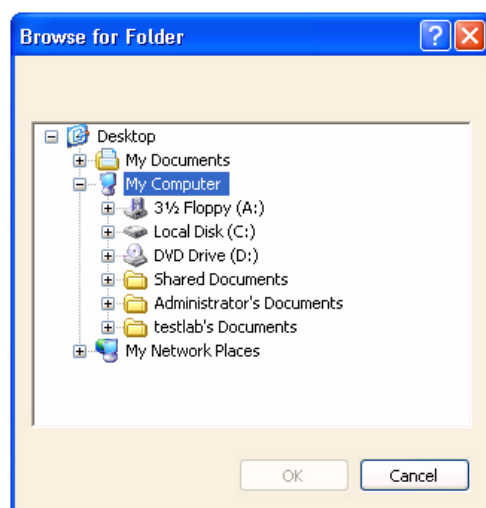


Task: *To add an existing folder to a Citrix profile:*

1. In the **Citrix Assistant**, open the **Profile Files** page. The files and folders are listed in the **Citrix Profile** tree, organized by installation directory.

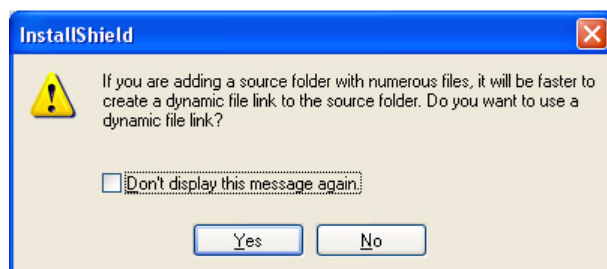


2. Browse through the folder tree to find the folder that you would like to add a folder into.
3. Select the folder and click the **Add Folders** button. The **Browse for Folder** dialog box opens, listing all of the directories available to your computer.



4. Select a folder and click **OK**.

If you are editing an InstallShield project (not a Windows Installer package), you are prompted to choose whether you want to create a dynamic file link to the source folder.



5. Indicate whether you want to create a dynamic file link by selecting one of the following:
 - **No**—For more flexibility with Citrix options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
 - **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See Dynamic File Link Settings Dialog Box.

The folder that you selected is now listed, along with of the files and folders within it.

Creating a New Folder

You can create a new, empty folder by selecting an existing folder in the tree and selecting **New Folder** from the context menu.



Task: To create a new folder:

1. Right-click on a folder in the **Citrix Profile** tree and select **New Folder**. A new folder is created as a subfolder of the selected folder:



2. Enter a name for the new folder.

Moving Files and Folders

To change the folder's location in the Citrix Profile folder tree structure, perform the following steps:



Task: *To move a file or folder:*

1. Select the file or folder that you want to move.
2. With the mouse button down, drag the file or folder to the new location.
3. Release the mouse button.

Deleting Files and Folders

To delete a file or a folder (and all of its contents) from a Citrix profile, perform the following steps:



Task: *To delete a file or folder:*

1. Select the file or folder in the **Citrix Profile** tree that you want to delete.
2. Select **Delete** from the context menu. You are prompted to confirm the deletion.
3. Click **Yes**. The selected file or folder is deleted.



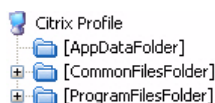
Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains from the entire Project, not just from the Citrix profile.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).

Controlling the Display of Predefined Folders

On the **Profile Files** page, the **Citrix Profile** tree initially displays the more commonly used predefined folders, such as [ProgramFilesFolder] and [CommonFilesFolder].



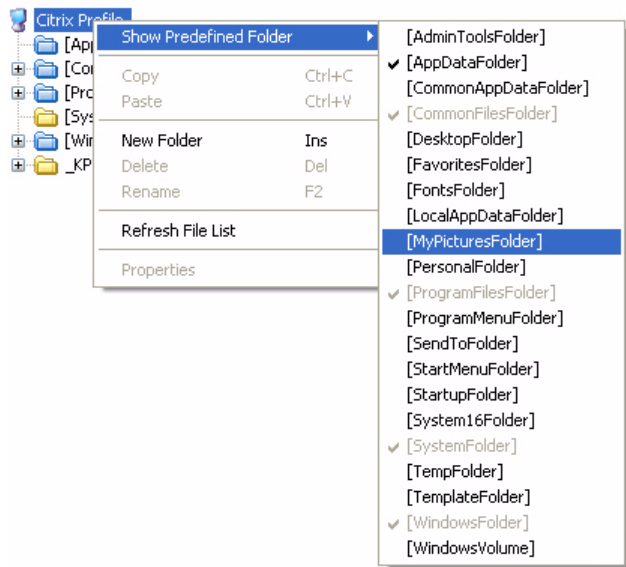
These predefined folders are dynamic, meaning that they do not use hard-coded paths. The value for each destination folder is obtained from the operating system of the target machine.

You can control which predefined folders are listed in this tree.



Task: *To change which predefined folders are listed:*

1. In the **Citrix Profile** tree, select the **Citrix Profile** node (or any of the files or folders that are listed, point to **Show Predefined Folder**. A list of predefined folders opens.



Those folders that are already displayed are preceded by a check mark, and those that are not displayed do not have a check mark.

2. To add a folder to the tree listing, select a folder that is not currently listed in the tree.



Note • These predefined folders are always added to the root of the Citrix Profile tree, no matter what file or folder you had selected when you selected it from the Predefined Folders list.

3. To remove a folder from the tree listing, select that folder name in this list (which is preceded by a check mark).



Note • You cannot turn off the display of the [ProgramFilesFolder].

Setting Isolation Options

The Citrix XenApp uses isolation environments to control application compatibility and accessibility. The isolation option that is assigned to a file, folder or registry key specifies how the isolation environment will provide access to system resources requested by the application.

The default settings for isolation options are set on the Citrix XenApp, and those defaults are adequate for most environments. However, you can override the default settings for selected files, folders, or registry keys to exert control over application interactions with client operating system resources.

You set isolation options on the **Isolation Options** dialog box, which is open by selecting a file or folder and then selecting **Isolation Options** from the context menu.

Information about setting isolation options is presented in the following topics:

- [Overview of Citrix Isolation Options](#)
- [Setting Isolation Options for Folders and Files](#)
- [Inheritance of Isolation Options from Folders to Files](#)



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.*

Overview of Citrix Isolation Options

The Citrix XenApp uses isolation environments to control application compatibility and accessibility. The isolation option that is assigned to a file, folder or registry key specifies how the isolation environment will provide access to system resources requested by the application.

The default settings for isolation environments are set on the Citrix XenApp, and those defaults are adequate for most environments. However, in the Citrix Assistant, you can override the default settings for selected files, folders, or registry keys to exert control over application interactions with client operating system resources.

You set isolation options on the **Isolation Options** dialog box, which is opened by selecting **Isolation Options** on the context menu when you have a file or folder selected on the **Profile Files** page or a registry key selected on the **Profile Registry** page.

On the **Isolation Options** dialog box, you can choose one of the following isolation options:

Table 10-53 • Isolation Options


Option	Description
Default	<div>Select this option if you want the default isolation option for this file/folder/registry key as defined on the Citrix XenApp to be applied to this selection. This is the default selection for all files, folders, and registry keys.</div> <div> Caution • <i>You should select this option unless you require specific custom handling.</i></div>

Table 10-53 • Isolation Options (cont.)

Option	Description
Ignore	<p>Choose the Ignore option to direct the isolation environment to <i>always</i> use the copy of this selected file/folder/registry key that is on the system, not the one inside the isolation environment.</p> <p>Choosing this option gives the isolation environment direct access to the same location on the system that a non-isolated version of this application would have. By assigning the Ignore isolation option, you are creating a “hole” in the isolation environment to allow an application to write to the underlying system.</p> <p>For example, you would select Ignore in the following situations:</p> <ul style="list-style-type: none"> • If an application creates a directory for per-user data that is stored in a non-standard location. • If the workstation has extra drive volumes and an installer writes to those drives while installing into a target. • If your file share volume is on your packaging workstation. • When the Citrix profile needs to share data with an application outside the isolation environment, such as when users print to a network printer.
Isolate	<p>Choose the Isolate option to direct the isolation environment to first try to find the copy of this file/folder/registry key that is inside the isolation environment. If the item is not found there, then the isolation environment will use the copy of this file/folder/registry key that is on the system. Selecting Isolate ensures that the isolation environment is not given direct write access to the specified system resource.</p>
Strictly Isolate	<p>Choose the Strictly Isolate option to direct the isolation environment to always use the copy of this file/folder/registry key that is in the isolation environment, not on the system. This is useful when running two versions of an application on the same machine.</p>
Redirect	<p>Choose the Redirect option to redirect a request by the isolation environment for a file/folder/registry key to a specified location on the system (without first searching the user profile root and installation root locations).</p> <p>When selecting this option, you also need to select the location that the isolation environment should redirect to:</p> <ul style="list-style-type: none"> • Source—Lists the name of the selected item (filename, folder name, registry key). • Destination—<i>[Files and folders only]</i> Click the Browse [...] button and select the file or folder on the system that you want to redirect to. • Destination Root—<i>[Registry keys only]</i> Select the registry root of the registry key on the system that you want to redirect to. • Destination Key—<i>[Registry keys only]</i> Select the registry key on the system that you want to redirect to.

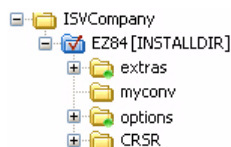
Setting Isolation Options for Folders and Files

To override a file or folder's default isolation options set on the Citrix XenApp, perform the following steps:



Task: *To set an isolation option on a folder or file.*

1. Open the **Profile Files** page.
2. Browse through the folder tree to find the file or folder that you would like to modify.
3. Select the file or folder and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.
4. Select one of the following options, as described in [Overview of Citrix Isolation Options](#).
 - [Default](#)
 - [Ignore](#)
 - [Isolate](#)
 - [Strictly Isolate](#)
 - [Redirect](#)
5. Click **OK**. Files and folders that have an isolation setting other than default are marked with a special icon:



Inheritance of Isolation Options from Folders to Files

Isolation options for files, folders and registry keys are always inherited. The Citrix isolation environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for C:\Windows and one for C:\Windows\System32. When the application requests C:\Windows\System32\notepad.exe, then the C:\Windows\System32 isolation rule will be applied because C:\Windows\System32 is a more specific reference to C:\Windows\System32\notepad.exe than is C:\Windows.

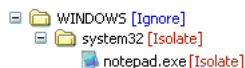


Figure 10-52: Example of Inheritance of Isolation Options from Folders to Files

Modifying Profile Shortcut Settings

You define profile shortcuts to enable users to launch a Citrix profile from within the isolation environment.

By default, the **Citrix Assistant** creates shortcuts to all of the executable (.exe) files that were added to the profile on the **Profile Files** page. These shortcuts are listed in a checklist on the **Profile Shortcuts** page.



Tip • Citrix currently only supports 16 color icons for shortcuts. Therefore, if you specify an **Icon File** on the **Shortcuts** view of the *Installation Designer*, be sure to select an icon that includes only 16 colors.

When you select each shortcut, details about it are displayed:

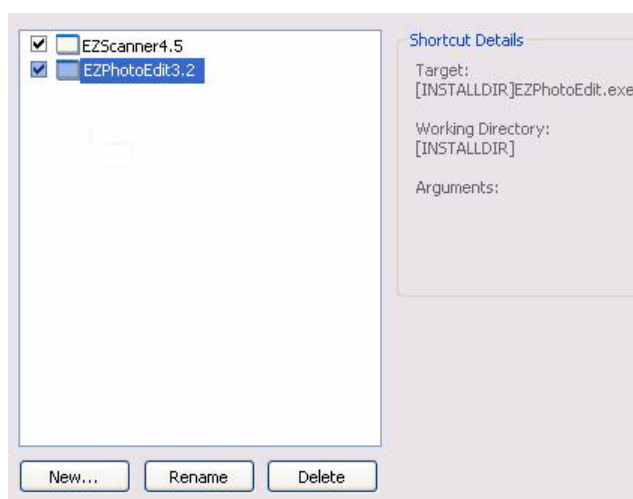


Figure 10-53: Initial List of Shortcuts for an Application



Caution • You must define at least one shortcut to enable users to launch the application from the isolation environment.

On the **Profile Shortcuts** page, you can create, delete, include, exclude, or rename a profile's shortcuts.

- [Shortcuts and the Isolation Environment](#)
- [Shortcut Requirements](#)
- [Creating a New Profile Shortcut](#)
- [Including an Existing Profile Shortcut](#)
- [Excluding vs. Deleting a Profile Shortcut](#)
- [Conditions When a Shortcut Should be Excluded or Deleted](#)
- [Renaming a Shortcut](#)

Shortcuts and the Isolation Environment

When a profile is published on the Citrix XenApp, the administrator has the option of placing available shortcuts on the client's desktop, client's Start menu, or only in the Citrix Program Neighborhood Agent applications list.

Shortcut presentation is specified in the **Application shortcut placement** area of the **Shortcut presentation** view of the Citrix Access Management Console **Application Properties** dialog box.

In the **Application shortcut placement** area, you have the following options:

Table 10-54 • Shortcut Presentation Options

Option	Description
Add to the client's Start menu	<p>Select this option to create a shortcut to this application in the user's local Start menu. A Client Application folder appears in the first pane of the Start menu:</p> <pre> Start MyApplicationFolder ApplicationName </pre> <p>When you select this option, the Place under Programs folder and Start menu folder fields are enabled.</p>
<ul style="list-style-type: none"> • Place under Programs folder (Program Neighborhood Agent Only) • Start menu folder (Program Neighborhood Agent Only) 	<p>Select the Place under Programs folder option to create a shortcut to this application under the Programs folder of the user's local Start menu.</p> <ul style="list-style-type: none"> • If you leave the Start menu folder field blank, the shortcut is created in root folder of the Programs menu. <pre> Start Programs MyApplicationFolder ApplicationName </pre> • If you specify a folder structure in the Start menu folder field, the shortcut is created in that folder structure within the local Programs folder, with each folder name separated with a backslash. For example, if you entered the following in the Start menu folder field: <pre>MyApplicationFolder/ApplicationTools</pre> <p>Then, the shortcut would be created in the following folder structure:</p> <pre> Start Programs MyApplicationFolder ApplicationTools ApplicationName </pre>

Table 10-54 • Shortcut Presentation Options (cont.)

Option	Description
Add shortcut to the client's desktop	Select this option to create a shortcut to this application on the user's local desktop.

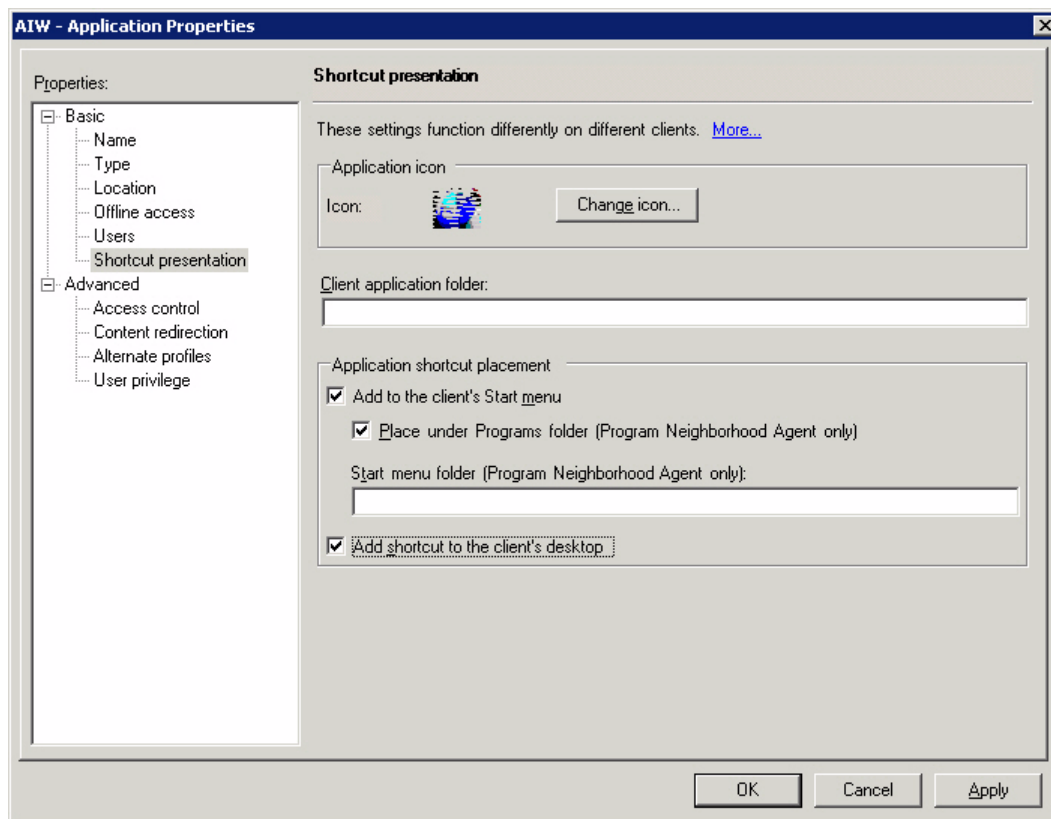


Figure 10-54: Shortcut Presentation View of the Citrix Presentation Server Application Properties Dialog Box

Shortcut Requirements

For each Citrix profile, you are required to define **at least one** shortcut. Profile shortcuts enable users to access the isolation environment and launch the application. If you build a Citrix profile that does not contain any shortcuts, users will not be able to launch the application.

Creating a New Profile Shortcut

On the **Profile Shortcuts** page, you can add a new shortcut to a file within the Citrix profile.



Task:

To create a new shortcut:

1. Open the **Profile Shortcuts** page. All of the shortcuts are listed:
 - Those that are currently included in the profile are selected.
 - Those that are currently excluded from the profile are not selected.
2. Click **New**. The **Browse for a Shortcut Target File** dialog box opens and prompts you to select a file within this profile.
3. Select the file that you want to create a shortcut to.
4. Click **Open**. A new shortcut is listed, and it is named the same name as the selected file.
5. To include this shortcut in the Citrix profile, make sure that its check box is selected.

Including an Existing Profile Shortcut

If you want to include a previously excluded shortcut in a Citrix profile, perform the following steps:



Task:

To include an existing profile shortcut:

1. Open the **Profile Shortcuts** page. All of the executable (.exe) files that were added on the **Profile Files** page are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To include a previously excluded shortcut, select the shortcut and select the check box.

Excluding vs. Deleting a Profile Shortcut

By default, the **Citrix Assistant** creates shortcuts to all of the executable (.exe) files that were added on the **Profile Files** page, and lists them in a checklist on the **Profile Shortcuts** page.

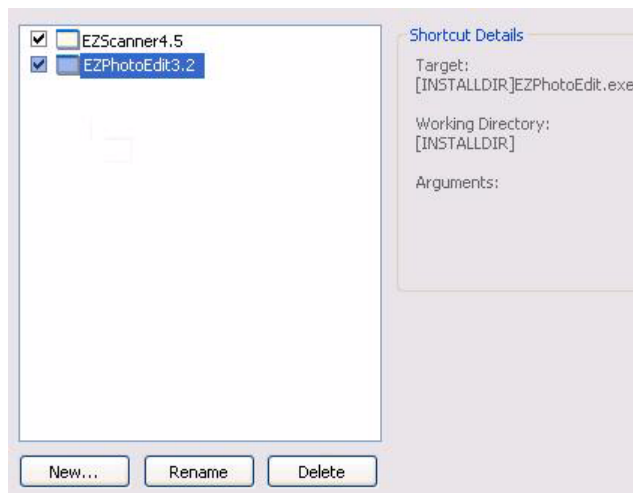


Figure 10-55: Initial List of Shortcuts for an Application

To prevent the shortcut from being created in the Citrix profile, you can choose to either delete or exclude it.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the Citrix profile, but it will remain in the InstallShield project. See [Excluding a Profile Shortcut](#).
- **Deleting a Shortcut**—When you delete a shortcut, it is removed from both the Citrix profile and the InstallShield project. See [Deleting a Shortcut](#).

If you have any unnecessary shortcuts in your project, you can simply exclude them from the Profile by unchecking them in the shortcuts list. If you like to permanently remove a shortcut, you can delete it from the shortcut list.

Excluding a Profile Shortcut

If you want to exclude one of these shortcuts from being created in the Citrix profile, perform the following steps:



Task: *To exclude a shortcut:*

1. Open the **Profile Shortcuts** page. All of the executable (.exe) files that were added on the **Profile Files** page are listed.
 - Those that are currently included are selected.
 - Those that are currently excluded are not selected.
2. To exclude a shortcut, select the shortcut and clear the check box.



Note • When you exclude a shortcut, it will not be created in the Citrix profile, but it will remain in the InstallShield project.

Deleting a Shortcut

To delete a shortcut, perform the following steps.



Task: To delete a shortcut:

1. Open the **Profile Shortcuts** page. All of the shortcuts are listed.
2. Select the shortcut and click **Delete**.



Note • If you delete a shortcut on the **Profile Shortcuts** page, the shortcut is also deleted from the InstallShield project, and, subsequently, from the Windows Installer package.

Conditions When a Shortcut Should be Excluded or Deleted

To prevent a shortcut from being created in the Citrix profile, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the Citrix profile, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding a Profile Shortcut](#).
- **Deleting a Shortcut**—When you delete a shortcut, it is removed from both the Citrix profile and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting a Shortcut](#).

Renaming a Shortcut

To rename a shortcut, perform the following steps:



Task: To add or delete a shortcut:

1. Open the **Profile Shortcuts** page. All of the executable (.exe) files that were added on the **Profile Files** page are listed.
2. Select the shortcut that you want to rename and click **Rename**. A box appears around the shortcut name, and the shortcut name becomes an editable field.
3. Enter a new name for the shortcut.

Modifying Profile Registry Settings

Using the **Citrix Assistant**, you can add, delete, or modify the registry settings in your Citrix profile.

You can also override the Citrix default isolation options for selected registry keys. Isolation options specify how the isolation environment will provide access to system resources requested by the application.

Information about modifying profile registry settings on the **Profile Registry** page includes the following topics:

- [About the Windows Registry](#)
- [Adding or Deleting Registry Keys and Values](#)
- [Setting Registry Isolation Options](#)

About the Windows Registry

The Windows registry is a system-wide database that contains configuration information used by applications and the operating system. The registry stores all kinds of information, including the following:

- Application information such as company name, product name, and version number
- Path information that enables your application to run
- Uninstallation information that enables end users to uninstall the application easily without interfering with other applications on the system
- System-wide file associations for documents created by an application
- License information
- Default settings for application options such as window positions

Keys, Value Names, and Values

The registry consists of a set of keys arranged hierarchically under the My Computer explorer. Just under My Computer are several root keys. An installation can add keys and values to any root key of the registry. The root keys that are typically affected by installations are:

- HKEY_LOCAL_MACHINE
- HKEY_USERS
- HKEY_CURRENT_USER
- HKEY_CLASSES_ROOT

A key is a named location in the registry. A key can contain a subkey, a value name and value pair, and a default (unnamed) value. A value name and value pair is a two-part data structure under a key. The value name identifies a value for storage under a key, and the value is the actual data associated with a value name. When a value name is unspecified for a value, that value is the default value for that key. Each key can have only one default (unnamed) value.

Note that the terms key and subkey are relative. In the registry, a key that is below another key can be referred to as a subkey or as a key, depending on how you want to refer to it relative to another key in the registry hierarchy.

Adding or Deleting Registry Keys and Values

Editing the registry on the **Profile Registry** page is performed much like it is performed on the InstallShield **Registry View**. See [Editing the Registry](#).

Setting Registry Isolation Options

To override a registry key's default isolation options set on the Citrix XenApp, perform the following steps:



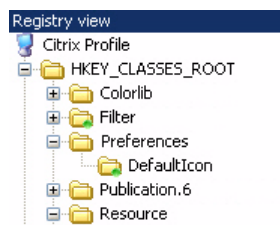
Task: *To set an isolation option on a registry key:*

1. Open the **Profile Files** page.
2. Browse through the registry tree to find the key that you would like to modify.
3. Select the folder or key and click **Isolation Options** on the context menu. The **Isolation Options** dialog box opens.



Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

4. Select one of the following options, as described in [Overview of Citrix Isolation Options](#).
 - [Default](#)
 - [Ignore](#)
 - [Isolate](#)
 - [Strictly Isolate](#)
 - [Redirect](#)
5. Click **OK**. Registry keys that have an isolation setting other than default are marked with a special icon:



Tip • To import an existing registry (.reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.

Inheritance of Isolation Options in the Registry

Isolation options for registry keys are always inherited. The Citrix isolation environment will apply the most specific reference to that resource.

For example, suppose you have an isolation option for the `Microsoft` registry key and one for `Microsoft\Windows` registry key. When the application requests `Microsoft\Windows\CurrentVersion`, then the `Microsoft\Windows` isolation rule will be applied because `Microsoft\Windows` is a more specific reference to `Microsoft\Windows\CurrentVersion` than is `Microsoft`.

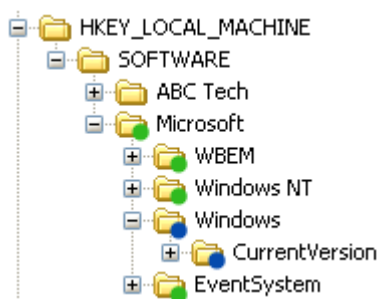


Figure 10-56: Example of Inheritance of Isolation Options from Folders to Files

Modifying Build Settings

On the **Build Settings** page, you choose which releases of this InstallShield project you want to build a Citrix profile for when the project is built, specify whether you want to digitally sign the Citrix profile, and specify whether you want to include additional Windows Installer packages in the Citrix profile.

Also, if you are editing a Windows Installer package in Direct Edit mode (or Direct MST mode), you need to select the **Build Citrix Profile** option on the **Build Settings** page before you will be able to build a Citrix profile for that Windows Installer package.

- [Selecting Releases to Build](#)
- [Digitally Signing a Citrix Profile](#)
- [Including Additional Windows Installer Packages in a Citrix Profile](#)
- [Enabling Citrix Profile Building When in Direct Edit Mode](#)



Important • You must create at least one Release (on the **Releases** view of the Installation Designer) before you will be able to select a Release on the **Build Settings** page.

Selecting Releases to Build

You select the releases that you want to build a Citrix profile for on the **Releases** tree of the **Build Settings** page.



Important • You cannot create or edit a release in the Citrix Assistant. If no releases exist, you can simply click the **Build** toolbar button to create a new release or open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build a Citrix profile. For more information, see [Creating and Building Releases](#).

If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Settings** page is not displayed.



Task: To select releases to build:

1. Open the **Build Settings** page.
2. Select the releases in the **Releases** tree that you want to build a Citrix profile for.



Important • When you select a release on the **Build Settings** page, you are specifying that whenever you build that particular release, you want to also build a Citrix profile for that release. However, the releases that are selected on the **Build Settings** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Settings** page:

- **Active release selected**—A Windows Installer package and a Citrix profile would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See [Performing Batch Builds](#).

Digitally Signing a Citrix Profile

You can digitally sign your Citrix profile to assure end users that neither your installation nor the code within your application has been tampered with or altered since publication. When you digitally sign your application, end users are presented with a digital certificate when they run your installation.



Task: *To digitally sign a Citrix profile:*

1. Open the **Build Settings** page.
2. Select the **Digitally sign Citrix profile** option. The **Personal Information Exchange file (.pfx)** field is enabled. A .pfx file is a standard file format for digital certificates.
3. Click **Browse** and select the .pfx file that you want to use to digitally sign this Citrix profile.





Including Additional Windows Installer Packages in a Citrix Profile

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual application, you can also use the Citrix Assistant to convert an application suite of multiple Windows Installer packages into one virtual application.

To include additional Windows Installer packages in a Citrix profile, set the **Would you like to include additional MSI files in the virtual package?** option on the **Build Settings** page to **Yes**, and then select the packages that you want to add.



Task: *To include additional Windows installer packages in a Citrix profile:*

1. Open the **Build Settings** page.
2. Set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**.
3. Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.
 - The order of the packages can be changed by selecting a package in the list and clicking the Move Up () and Move Down () buttons.
 - Use the Delete button () to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Enabling Citrix Profile Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **Citrix Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build a Citrix profile for this Windows Installer package. To do this, perform the following steps:



Task: *To enable Citrix profile building when in Direct Edit Mode:*

1. Open a Windows Installer package or a transform file in InstallShield. It will be opened in Direct Edit Mode or Direct MST Mode, and the Build function (**Build** on the **Build** menu and the **Build** toolbar button) will be disabled.
2. Open the **Build Settings** page of the Citrix Assistant.
3. Select the **Build Citrix Profile** option. After you select this option, the **Build Citrix Profile** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.

Building a Citrix Profile

The method for building a Citrix profile depends upon what file you have open—an InstallShield project or a Windows Installer package.

- [Building a Citrix Profile for an InstallShield Project](#)
- [Building a Citrix Profile for a Windows Installer Package](#)

Building a Citrix Profile for an InstallShield Project

To build a Citrix profile for an InstallShield project, perform the following steps:



Task: *To build a Citrix profile for an InstallShield project:*

1. Open the InstallShield project in InstallShield.
2. On the **Releases** view of the Installation Designer, make sure that at least one release has been created, and select the release that you want to build.



Important • You cannot create or edit a release in the Citrix Assistant. If no releases exist, or if you want to create a new release, open the **Releases** view of the InstallShield Installation Designer. You must create at least one release before you will be able to build a Citrix profile. For more information, see [Creating and Building Releases](#).

3. Open the **Build Settings** page of the Citrix Assistant.

4. In the **Releases** tree, select the same release that is selected on the **Releases** view of the InstallShield Installation Designer. This is the release that you will build a Citrix profile for.



Important • When you select a release on the **Build Settings** page, you are specifying that whenever you build that particular release, you want to also build a Citrix profile for that release. However, the releases that are selected on the **Build Settings** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what was selected on the **Build Settings** page:

- **Active release selected**—A Windows Installer package and a Citrix profile would be built.
- **Active release not selected**—Only a Windows Installer package would be built.

To build more than one release at a time, perform a batch build. See *Performing Batch Builds*.

5. Click the **Build** toolbar button (or select **Build Release** on the **Build** menu) to start building the active release.

The output of the build will be a Windows Installer package and a Citrix profile. For a description of the files that comprise a Citrix profile, see [About Citrix Profiles](#).

Building a Citrix Profile for a Windows Installer Package

To build a Citrix profile for a Windows Installer package, perform the following steps:



Task: *To build a Citrix profile for a Windows Installer package:*

1. Do one of the following to open a Windows Installer package:
 - On the **File** menu, select **Open** and select a Windows Installer package (.msi).
 - On the **File** menu, select **Open** and select a transform file (.mst). The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
 - On the **File** menu, select **New** to open the **New Project** dialog box. Select **Transform** and click **OK**. The **Open Transform Wizard** opens and you are prompted to identify the transform file's associated Windows Installer package.
2. Use the Installation Designer to make any desired edits to the Windows Installer package or Transform file, and use the Citrix Assistant to set Citrix profile options.
3. On the **Build Settings** page of the Citrix Assistant, select the **Build Citrix profile** option.
4. Save the edits to the Windows Installer package or transform file by selecting **Save** on the **File** menu.
5. Click the **Build** toolbar button (or select **Build Citrix Profile** on the **Build** menu) to start building the Citrix profile.

The output of the build will be a Windows Installer package and a Citrix profile. For a description of the files that comprise a Citrix profile, see [About Citrix Profiles](#).



Note • For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see *Virtualization Conversion Errors and Warnings*.

Citrix Assistant Reference

Reference information about the Citrix Assistant is organized into the following sections:

- [Pages](#)
- [Dialog Boxes](#)
- [Building Citrix Profiles Using the Command Line](#)
- [Citrix Profile Conversion Error and Warning Messages](#)
- [Application Features Requiring Pre- or Post-Conversion Actions](#)

Pages

The Citrix Assistant is comprised of the following pages:

- [Home Page](#)
- [Profile Information Page](#)
- [Profile Requirements Page](#)
- [Profile Files Page](#)
- [Profile Shortcuts Page](#)
- [Profile Registry Page](#)
- [Build Settings Page](#)

Home Page

The Citrix Assistant Home page displays a diagram that illustrates the process of creating a Citrix profile for deployment on Citrix XenApp.

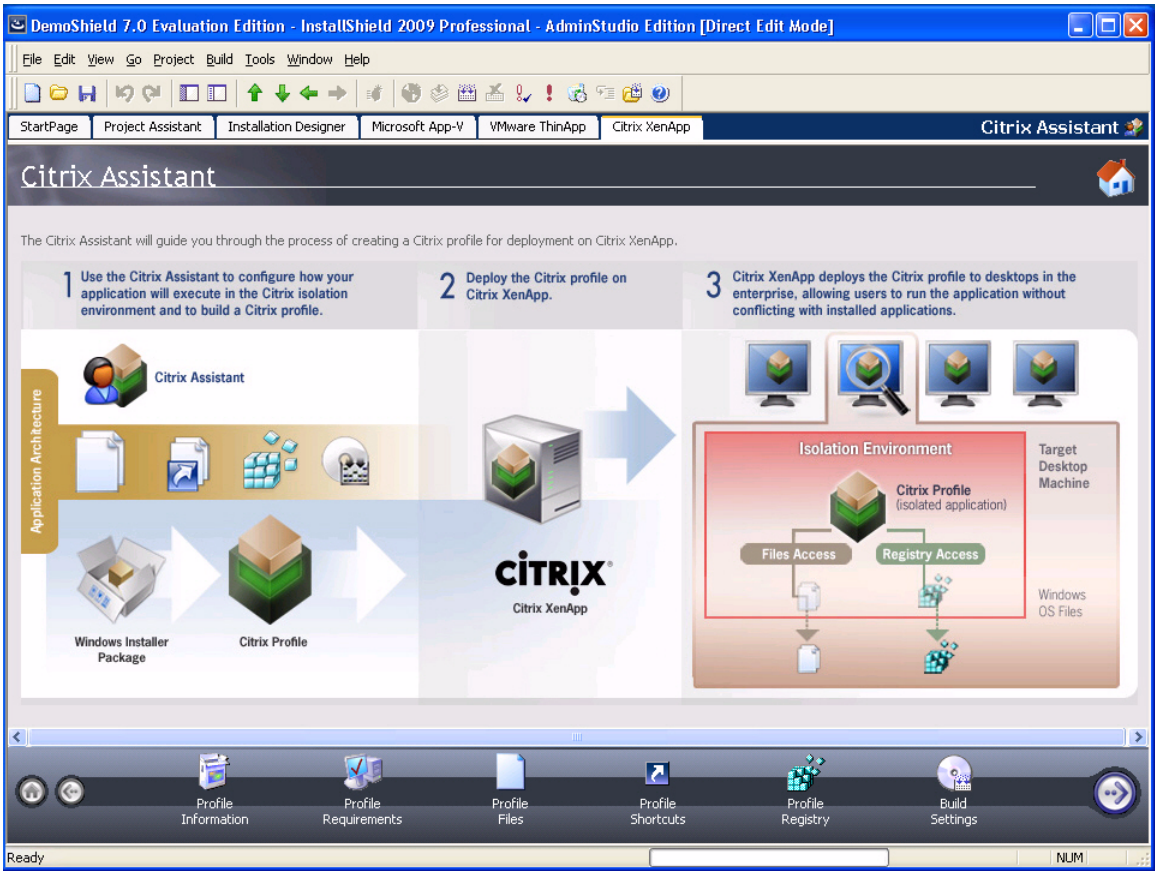









Figure 10-57: Citrix Assistant Home Page

Click the following icons in the navigation bar at the bottom of the page to navigate through the Citrix Assistant interface:

Table 10-55 • Navigation Bar Icons

Icon	Destination
	Profile Information Page
	Profile Requirements Page

Table 10-55 • Navigation Bar Icons

Icon	Destination
	Profile Files Page
	Profile Shortcuts Page
	Profile Registry Page
	Build Settings Page
	Go to next page.
	Jump back to previous page.
	Home Page

Profile Information Page

On the **Profile Information** page in the **Citrix Assistant**, you specify the **Name**, **Description**, and **Version** of the Citrix profile you are creating, and you specify the package **Security Settings** to determine whether this package can run executables that are not included with the Citrix profile.

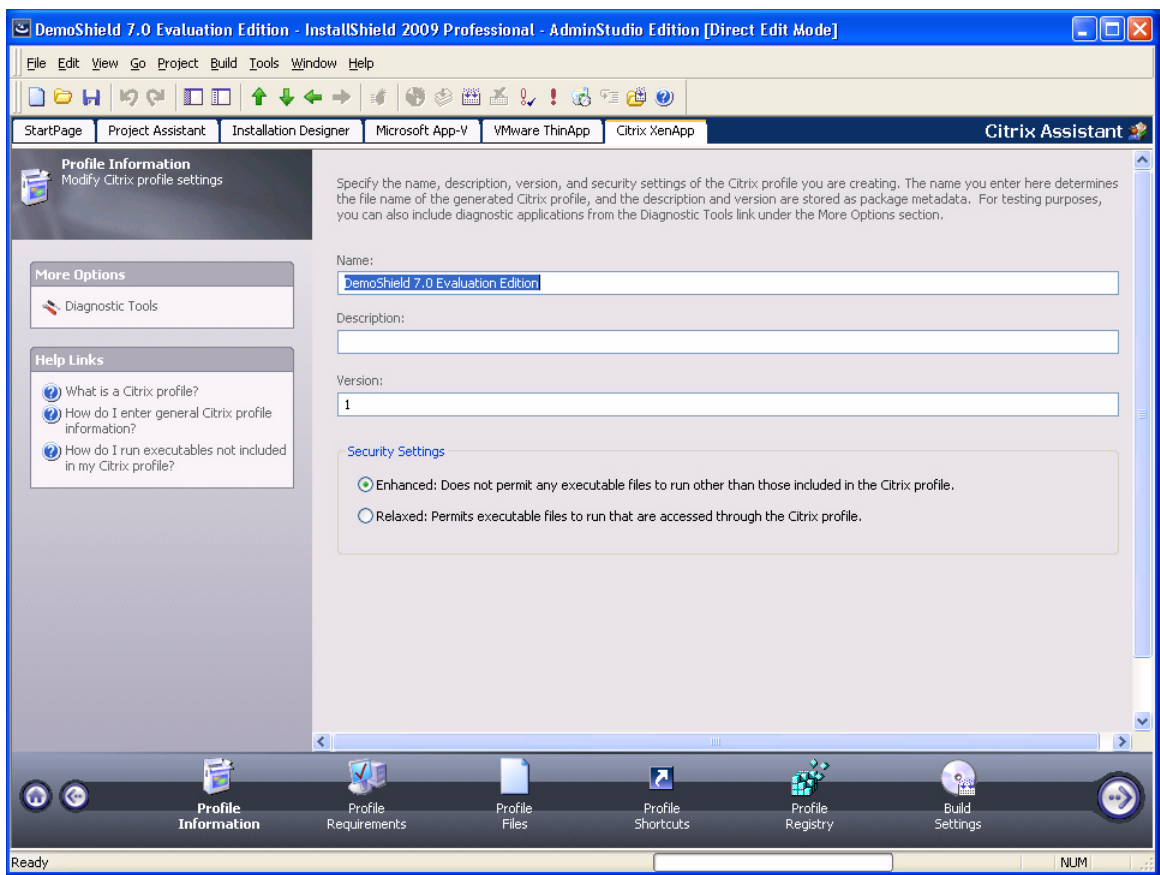


Figure 10-58: Citrix Assistant Profile Information Page

The Profile Information page includes the following options:

Table 10-56 • Profile Information Page


Option	Description
Name	<p>Enter a name for this Citrix profile. The name you enter here determines the file name of the generated Citrix profile.</p> <div></div> <p>Note • Do not include the version number in the package name.</p>
Description	<p>Briefly describe this package. This information is stored with the package as metadata.</p>
Version	<p>Enter the version number of this Citrix profile. This information is stored as package metadata.</p>

Table 10-56 • Profile Information Page (cont.)

Option	Description
Security Settings	<p>Select one of the following options:</p> <ul style="list-style-type: none">• Enhanced—Does not permit any executable files to run other than those included in the Citrix profile. For example, if the client is running an Internet Explorer plug-in included in the Citrix profile, the enhanced security setting prevents the client from running any other plug-ins that the user might download. <p>You would choose Enhanced security to disable an application from running executable files it writes to its current working directory on the client. Selecting Enhanced security might restrict or prevent the operation of some applications.</p> <ul style="list-style-type: none">• Relaxed—Permits executables files to run that are accessed through the Citrix profile. <p>You would choose Relaxed security to allow an application to run executable files it writes to its current working directory on the client. Choosing Relaxed security can enable the client to download and run potentially malicious software.</p>

For testing purposes, you can also choose to include diagnostic tools in your Citrix profile by selecting the **Diagnostic Tools** link in the **More Options** list. For more information, see [Diagnostic Tools Dialog Box](#).

Profile Requirements Page

On the **Profile Requirements** page of the Citrix Assistant, you can select the operating systems that client workstations must be running in order for your application to operate properly. By default, all operating systems that are supported by the Citrix client are selected.

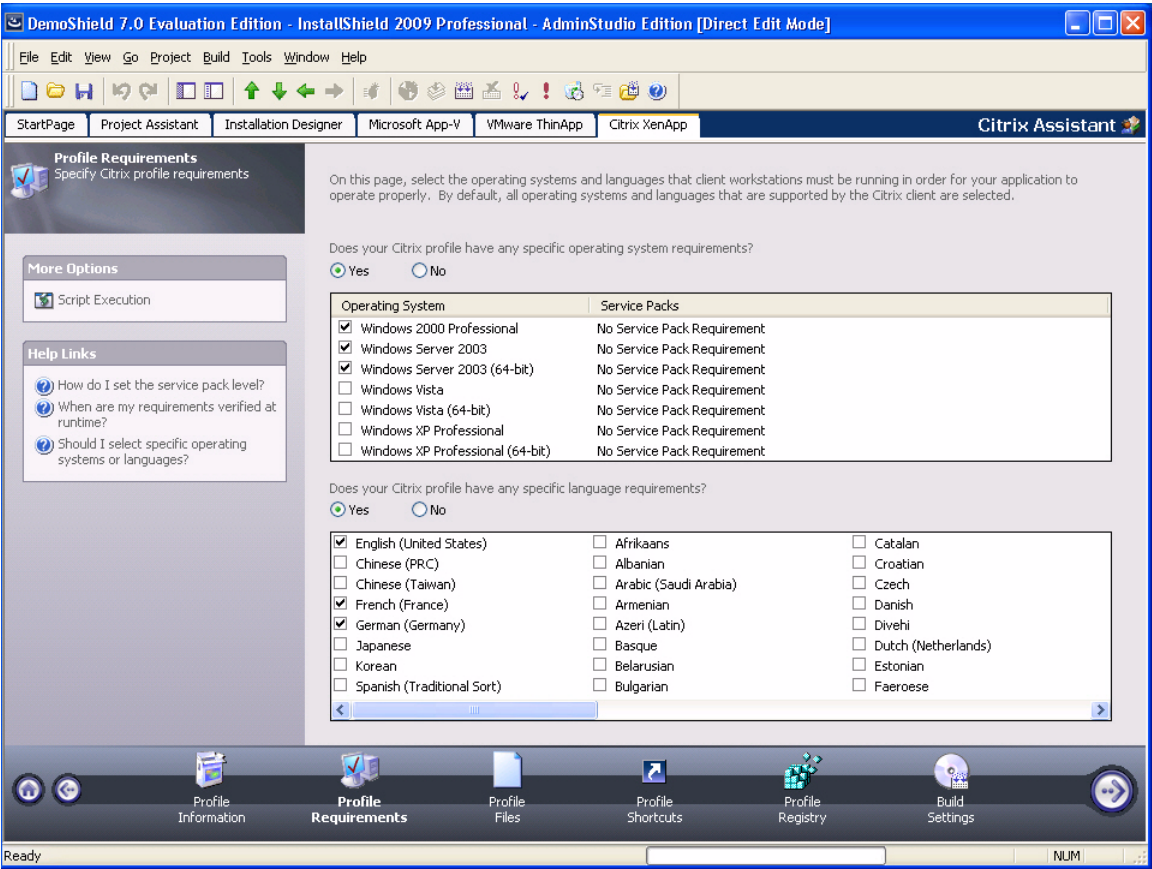


Figure 10-59: Citrix Assistant Profile Requirements Page

The **Profile Requirements** page has the following options:

Table 10-57 • Profile Requirements Page Options

Option	Description
Does your Citrix profile have any specific operating system requirements?	<p>Select one of the following:</p> <ul style="list-style-type: none">• Yes—Select this option if the application does not support one of the listed operating systems. When you select this option, the check boxes are unlocked and you can clear the selection of the unsupported operating systems.• No—Select this option if this application will run on all of the listed operating systems (which are the operating systems that the Citrix client supports). When this option is selected, the operating system check boxes are locked and cannot be changed.

Table 10-57 • Profile Requirements Page Options

Option	Description
Operating System / Service Packs List	<p>If you set the Does your Citrix profile have any specific operating system requirements? option to Yes, this list becomes enabled.</p> <p>To specify operating system requirements, first select the operating systems that this application supports, and clear those that this application does not support.</p> <p>Then, for each of the selected operating systems, right-click on it and select Service Packs Requirement from the context menu to open the Service Packs Requirements dialog box, and choose one of the following options:</p> <ul style="list-style-type: none"> • Not Required—This application supports all versions of this operating system, regardless of the number of Service Packs installed. • No Service Packs Allowed—This application only supports the initial release of this operating system; if any Service Packs are installed, this application will not run properly. • Exact Service Pack Level—This application requires the installation of a specific Service Pack on this operating system in order to run properly. Enter the required Service Pack Level in the box. • At least Service Pack Level—To run properly, this application requires that this operating system have at least the specified Service Pack (or higher) installed. Enter the minimum required Service Pack Level in the box. • At most Service Pack Level—To run properly, this application requires that this operating system have at most the specified Service Pack (or lower) installed. Enter the maximum required Service Pack Level in the box. • Range of Service Pack Levels—To run properly, this application requires that this operating system have a specified range of Service Packs installed. If you select this option, specify the Minimum Level and Maximum Level in the boxes.
Does your Citrix profile have any specific language requirements?	<p>Select one of the following:</p> <ul style="list-style-type: none"> • No—Select this option if this application will run on all of the listed languages (which are the languages that the Citrix client supports). When this option is selected, the language check boxes are locked and cannot be changed. • Yes—Select this option if the application does not support one of the listed languages. When you select this option, the check boxes are unlocked and only English is selected by default.

Table 10-57 • Profile Requirements Page Options

Option	Description
Language List	If you set the Does your Citrix profile have any specific language requirements? option to Yes , this list becomes enabled. Select only those languages that this application supports.

Profile Files Page

On the **Profile Files** page of the Citrix Assistant, you can perform the following tasks:

- [View Files and Folders](#)
- [Add Files and Folders](#)
- [Delete Files and Folders](#)
- [Set Isolation Options](#)
- [Modifying the Display of Predefined Folders](#)

View Files and Folders

On the **Profile Files** page, you can view all of the files and folders that are currently in your Citrix profile.

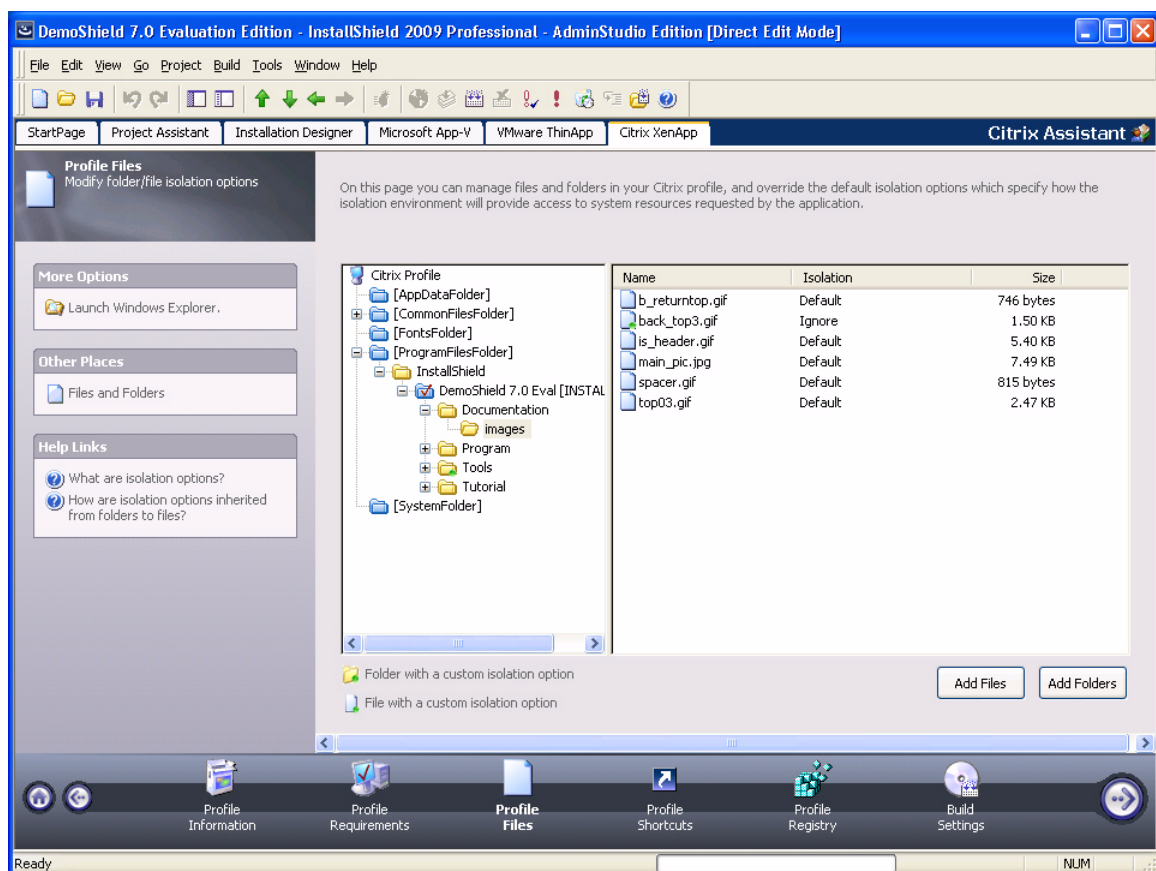


Figure 10-60: Citrix Assistant Profile Files Page

Folders are listed in the Citrix Profile tree on the left, and all of the files in the selected folder are listed on the right.

- The directories in the tree represent how your application will look when it is installed on to your customer's machine.
- Blue folders are the supported MSI standard folders.
- The folder with the check mark is INSTALLDIR, which represents the main product installation directory.

Add Files and Folders

On the **Profile Files** page, you can use the **Add Files** and **Add Folders** buttons to add new files and folders to include in the Citrix profile. See .

If you are editing an InstallShield project (not a Windows Installer package), and you are adding a folder to this profile, you are prompted to choose whether you want to create a dynamic file link to the source folder.

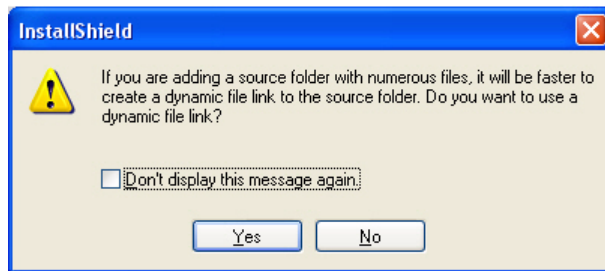


Figure 10-61: Dynamic File Link Option Dialog Box

Indicate whether you want to create a dynamic file link by selecting one of the following:

- **No**—For more flexibility with Citrix options, it is recommended that you select **No** to indicate that you *do not* want to use a dynamic file link, because you would then not be able to customize isolation options for any of the items in this folder.
- **Yes**—If you wish to use the default isolation options for all the files and folders under this folder, then select the dynamic file link option by clicking **Yes**. The **Dynamic File Link Settings** dialog box would then open, prompting you to specify the source folder for your dynamic link, and to set options regarding which files and folders to include in the dynamic link. See Dynamic File Link Settings Dialog Box.

Delete Files and Folders

You can delete files and folders from the Citrix profile by selecting the file or folder you want to delete, and selecting **Delete** from the context menu. For more information, see [Deleting Files and Folders](#).



Caution • If you choose to delete a folder, you are also deleting all of the files and subfolders that the folder contains.



Note • You cannot delete predefined folders. You can only turn off the display of those folders. For more information, see [Controlling the Display of Predefined Folders](#).



Tip • To select multiple files, use the Shift key (for contiguous files) or the Ctrl key (for non-contiguous files).

Set Isolation Options

The Citrix XenApp uses isolation environments to control application compatibility and accessibility. The isolation option that is assigned to a file, folder or registry key specifies how the isolation environment will provide access to system resources requested by the application.

The default settings for isolation environments are set on the Citrix XenApp, and those defaults are adequate for most environments. However, in the Citrix Assistant, you can override the default settings for selected files, folders, or registry keys to exert control over application interactions with client operating system resources.

You set isolation options by selecting a file or folder and then selecting **Isolation Options** from the context menu. For an overview of the available isolation options, and for instructions on how to set them, see [Setting ThinApp Isolation Options](#).

Modifying the Display of Predefined Folders

You can specify which of the Windows Installer predefined folders are listed in the **Citrix Profile** tree. See [Controlling the Display of Predefined Folders](#).

Profile Shortcuts Page

You define profile shortcuts to enable users to launch a Citrix profile from within the isolation environment.

By default, the **Citrix Assistant** creates shortcuts to all of the executable (.exe) files that were added to the profile. These shortcuts are listed in a checklist on the **Profile Shortcuts** page.



Note • Only shortcuts to executables are included in the profile. The Citrix Administrator chooses which of these shortcuts will be available to their users. When publishing, the Citrix Administrator chooses where to place the shortcut for their users to see.



Tip • Citrix currently only supports 16 color icons for shortcuts. Therefore, if you specify an **Icon File** on the **Shortcuts** view of the Installation Designer, be sure to select an icon that includes only 16 colors.

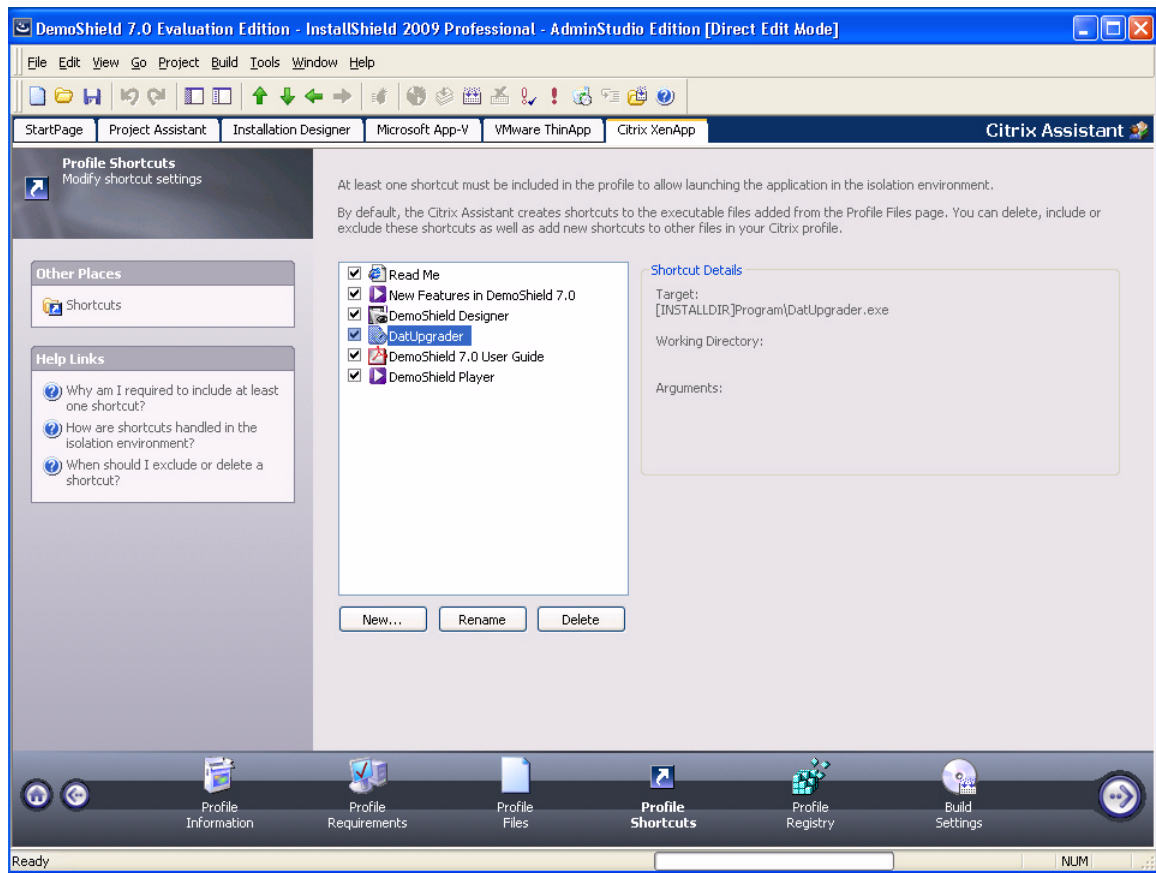


Figure 10-62: Profile Shortcuts Page

Shortcut Requirements

For each Citrix profile, you are required to define **at least one** shortcut. Profile shortcuts enable users to access the isolation environment and launch the application. If you build a Citrix profile that does not contain any shortcuts, users will not be able to launch the application.

Difference Between Deleting and Excluding a Profile Shortcut

To prevent a shortcut from being created in the Citrix profile, you can choose to either delete or exclude it, depending upon whether you want it to remain in the InstallShield project.

- **Excluding a shortcut**—When you exclude a shortcut, it will not be created in the Citrix profile, but it will remain in the InstallShield project. This means that the shortcut would be included in the Windows Installer package that is built from this InstallShield project. See [Excluding a Profile Shortcut](#).
- **Deleting a Shortcut**—When you delete a shortcut, it is removed from both the Citrix profile and the InstallShield project. This means that the shortcut would also be deleted from the Windows Installer package that is built from this InstallShield project. See [Deleting a Shortcut](#).

Managing Shortcuts

On the **Profile Shortcuts** page, you can create, delete, include, exclude, or rename a profile's shortcuts. For step-by-step instructions, see the following topics:

- [Creating a New Profile Shortcut](#)
- [Including an Existing Profile Shortcut](#)
- [Excluding vs. Deleting a Profile Shortcut](#)
- [Conditions When a Shortcut Should be Excluded or Deleted](#)
- [Renaming a Shortcut](#)

Profile Registry Page

On the **Profile Registry** page, you can view existing registry items, and add or delete registry items. You can also override the Citrix default isolation options for a registry key. Isolation options specify how the isolation environment will provide access to system resources requested by the application.

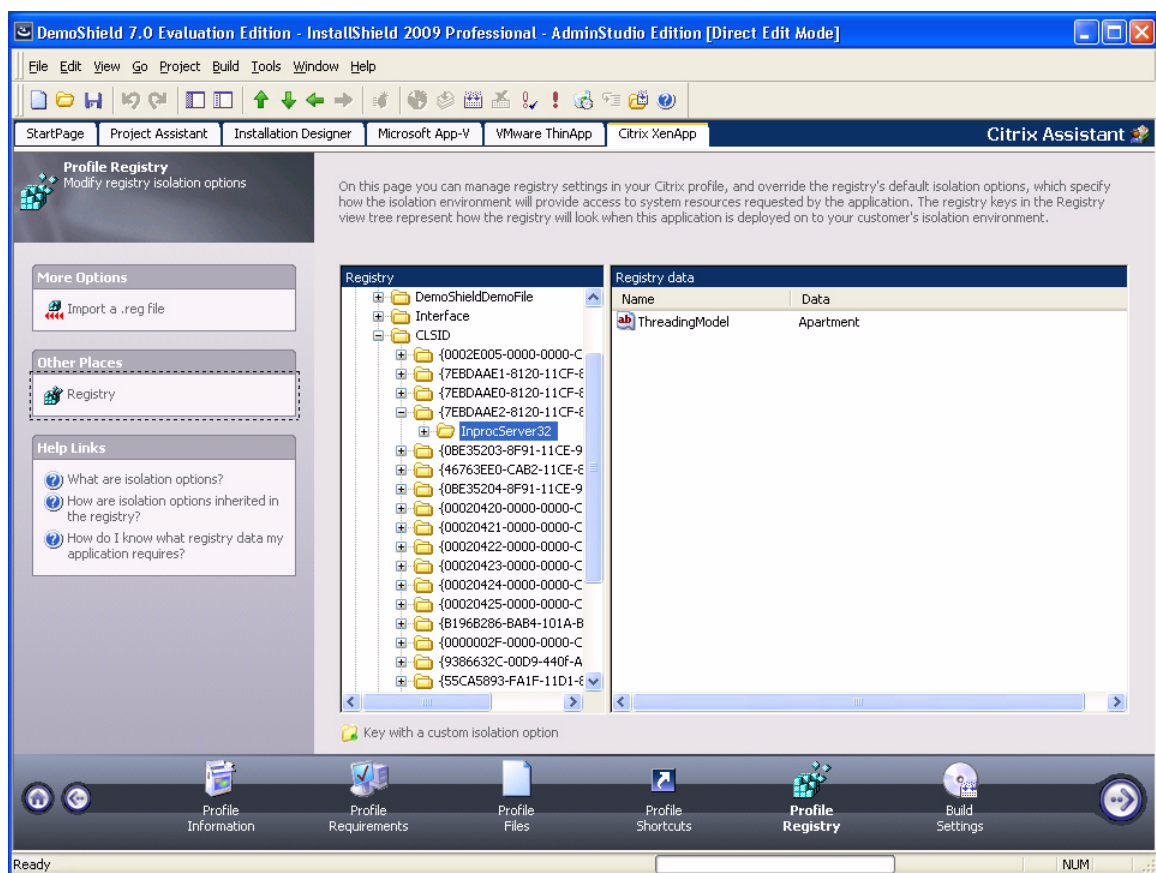


Figure 10-63: Citrix Assistant Profile Registry Page

Registry items that are listed on this page will be included in the Citrix profile, and those that you delete will not. By default, all new registry keys are isolated.



Tip • To import an existing registry (.reg) file, click the **Import a .reg file** option on the **More Options** list to open the Registry Import Wizard.



Note • You cannot set isolation options on root registry keys.

Editing the registry on the **Profile Registry** page is performed much like it is performed on the InstallShield **Registry View**. See [Editing the Registry](#).

For information on how to override a registry key's default isolation options, see [Setting Isolation Options](#).



Important • While you cannot explicitly set an isolation option on a registry value, registry values are subject to the isolation options of their keys.

Build Settings Page

On the **Build Settings** page, you can perform the following tasks:

- [Selecting Releases to Build](#)
- [Digitally Signing a Citrix Profile](#)
- [Including Additional Windows Installer Packages in a Citrix Profile](#)
- [Enabling Citrix Profile Building When in Direct Edit Mode](#)
- [Building a Citrix Profile](#)

Chapter 10: Creating Customized Virtual Applications

Creating Citrix Profiles

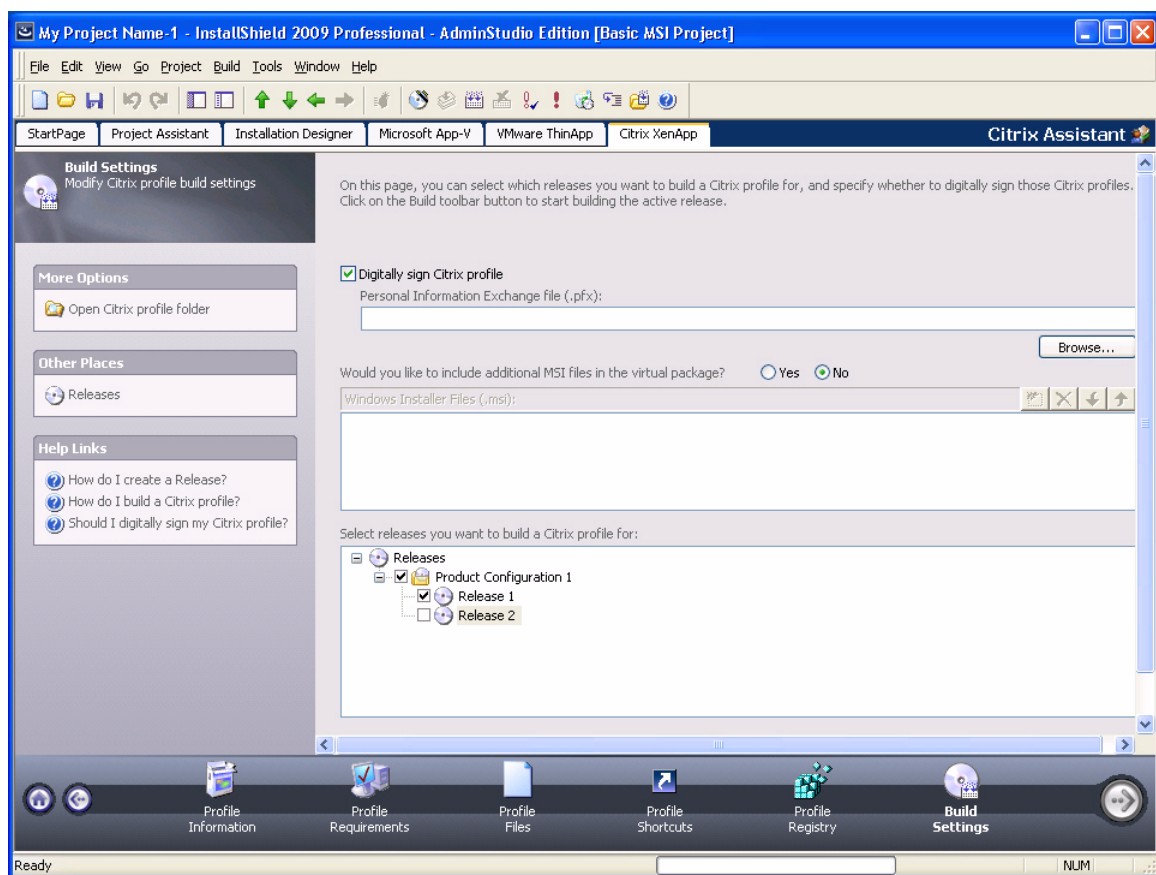


Figure 10-64: Citrix Assistant Build Settings Page

Selecting Releases to Build

You select the releases that you want to build a Citrix profile for on the **Releases** tree of the **Build Settings** page. By selecting a release, you are specifying that whenever that particular release is built, a Citrix profile will also be built.



Note • If you are editing a Windows Installer package (Direct Edit Mode) or transform file (Direct MST Mode), the **Releases** tree on the **Build Settings** page is not displayed.

About Building Releases

When you select a release on the Releases tree on the **Build Settings** page, you are specifying that whenever you build that particular release, you want to also build a Citrix profile for that release. However, the releases that are selected on the **Build Settings** page have no bearing upon which release is built when you click the **Build** button on the toolbar. When you initiate a build by clicking the **Build** button, a build is initiated for the **active** release—the release that was most recently selected on the Installation Designer **Releases** view. The output of that build would depend upon what releases were selected on the **Build Settings** page:

- **Active release selected**—A Windows Installer package and a Citrix profile would be built.
- **Active release not selected**—Only a Windows Installer package would be built.



Note • To build more than one release at a time, perform a batch build. See *Performing Batch Builds*.

About Creating Releases

You create and edit releases on the **Releases** view of the InstallShield Installation Designer. You cannot create or edit a release in the Citrix Assistant.

If no releases exist, or if you want to create a new release, open the **Releases** view of the Installation Designer. You must create at least one release before you will be able to build a Citrix profile. For more information, see [Creating and Building Releases](#).

Digitally Signing a Citrix Profile

You can digitally sign your Citrix profile to assure end users that neither your installation nor the code within your application has been tampered with or altered since publication. When you digitally sign your application, end users are presented with a digital certificate when they run your installation.

To digitally sign a Citrix profile, select the **Digitally sign Citrix profile** option on the **Build Settings** page. When this option is selected, the **Personal Information Exchange file (.pfx)** field is enabled. A .pfx file is a standard file format for digital certificates. You then click **Browse** and select the .pfx file that you want to use to digitally sign this Citrix profile.

Including Additional Windows Installer Packages in a Citrix Profile

Sometimes a primary Windows Installer package uses other Windows Installer packages indirectly, such as driver files, client components, etc. In addition to being able to convert a single Windows Installer package to a virtual package, you can also use the Citrix Assistant to convert an application suite of multiple Windows Installer packages into one virtual package.

To include additional Windows Installer packages in a Citrix profile, set the **Would you like to include additional MSI files in the virtual package?** option to **Yes**, and then select the packages that you want to add.

- Click the New button () and select the Windows Installer packages that you want to add. After each file is selected, it will be listed in the **Windows Installer Files (.msi)** list.

- The order of the packages can be changed by selecting a package in the list and clicking the Move Up (⬆) and Move Down (⬇) buttons.
- Use the Delete button (✖) to delete a package from the list.



Note • You can also convert a suite of applications into one virtual package using Windows Installer Converter. See [Converting a Suite of Applications into a Virtual Package](#).

Enabling Citrix Profile Building When in Direct Edit Mode

When you are editing a Windows Installer (.msi) package or a transform (.mst) file in the **Citrix Assistant**, you are in Direct Edit Mode or Direct MST Mode. Because you are directly editing a Windows Installer package, you save your changes by selecting **Save** on the **File** menu. It not necessary to build the package, because it is already built. Therefore, InstallShield's **Build** function is disabled.

However, you do need to run the build process to build a Citrix profile for this Windows Installer package. To enable the **Build** button to build the Citrix profile, select the **Build Citrix Profile** option on the **Build Settings** page.

After you select this option, the **Build Citrix Profile** selection on the **Build** menu becomes enabled, as does the **Build** toolbar button.

Building a Citrix Profile

The method for building a Citrix profile depends upon what file you have open—an InstallShield project or a Windows Installer package. For detailed instructions, see one of the following topics:

- [Building a Citrix Profile for an InstallShield Project](#)
- [Building a Citrix Profile for a Windows Installer Package](#)

Dialog Boxes

The Citrix Assistant includes the following dialog boxes:

- [Script Execution Dialog Box](#)
- [Diagnostic Tools Dialog Box](#)
- [File Isolation Options Dialog Box](#)
- [Folder Isolation Options Dialog Box](#)
- [Registry Isolation Options Dialog Box](#)
- [Service Packs Requirement Dialog Box](#)

Script Execution Dialog Box

On the **Script Execution** dialog box, which is opened by clicking **Script Execution** in the **More Options** list on the **Profile Requirements** page, you can choose to include scripts that must execute for your application to run properly. From this dialog box, you can view and manage all of the **Before Profile Launch** and **After Profile Exit** script files you are including with your Citrix profile.

- Files can be marked to run inside or outside of the isolation environment.
- Only files with .exe, .com, .cmd, or .bat extensions are allowed to execute.

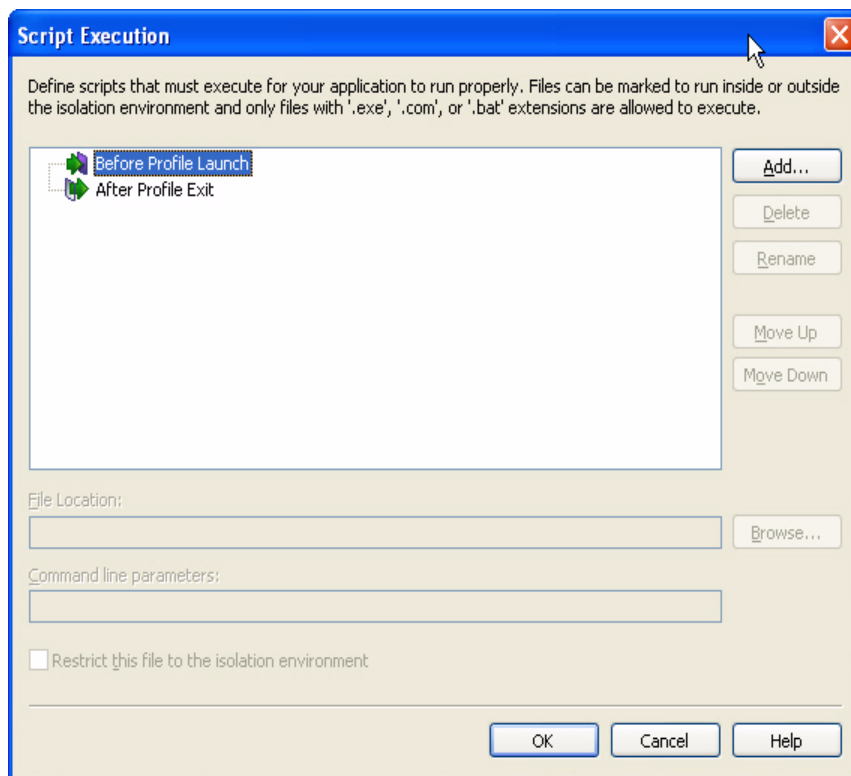


Figure 10-65: Custom Execution Dialog Box

For step-by-step instructions on using this dialog box, see [Adding Pre-Launch and Post-Exit Scripts](#).

Diagnostic Tools Dialog Box

On the **Diagnostic Tools** dialog box, which is opened by selecting **Diagnostic Tools** in the **More Options** list on the **Profile Information** page, you can choose to include the Windows Command Prompt and Registry Editor diagnostic tools with your Citrix profile.

If you include diagnostic tools with your Citrix profile, you will be able to look at the registry or file system for the application while it is running in its isolation environment. For example, if you were running a Citrix profile and got an error message stating that the application cannot load a DLL, you could use these diagnostic tools to troubleshoot the problem.



Caution • If you choose to include these diagnostic tools, the versions of `regedit.exe` and `cmd.exe` that are part of the operating system on the build machine are added to the Citrix profile. However, these tools may not be compatible with other operating systems.

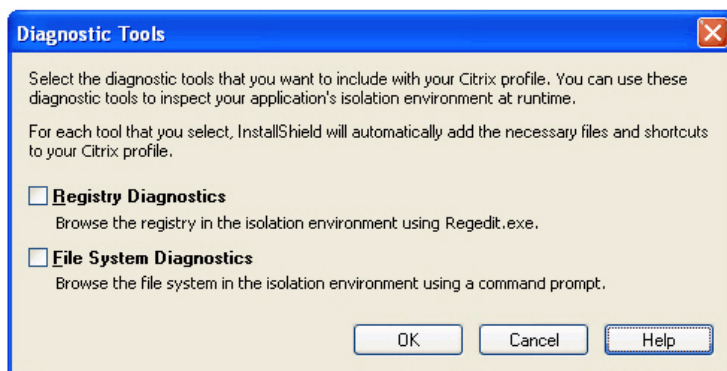


Figure 10-66: Diagnostic Tools Dialog Box

You can use these diagnostic tools to inspect your application's isolation environment at runtime. You have the following options:

Table 10-58 • Diagnostic Tools Dialog Box Options

Option	Description
Registry Diagnostics	Select this option if you want to include <code>regedit.exe</code> with your Citrix profile so that you can browse the profile registry.
File System Diagnostics	Select this option if you want to be able to browse the Citrix profile's isolation environment file system using a command prompt.

Launching the Diagnostic Tools Within the Isolation Environment

If you selected the **Registry Diagnostics** or **File System Diagnostics** options on the **Diagnostic Tools** dialog box, shortcuts to those tools are automatically added to the profile.

When the user runs this Citrix profile application, two additional shortcuts will be available in the application's shortcut folder: The names of these shortcuts will reflect the application name, such as:

```
[ProductName] Registry  
[ProductName] File System
```

When the user launches one of these shortcuts, that diagnostic tool is launched inside the context of the application's Citrix isolation environment.

File Isolation Options Dialog Box

On the **File Isolation Options** dialog box, you can override the default Citrix isolation option for the selected file.

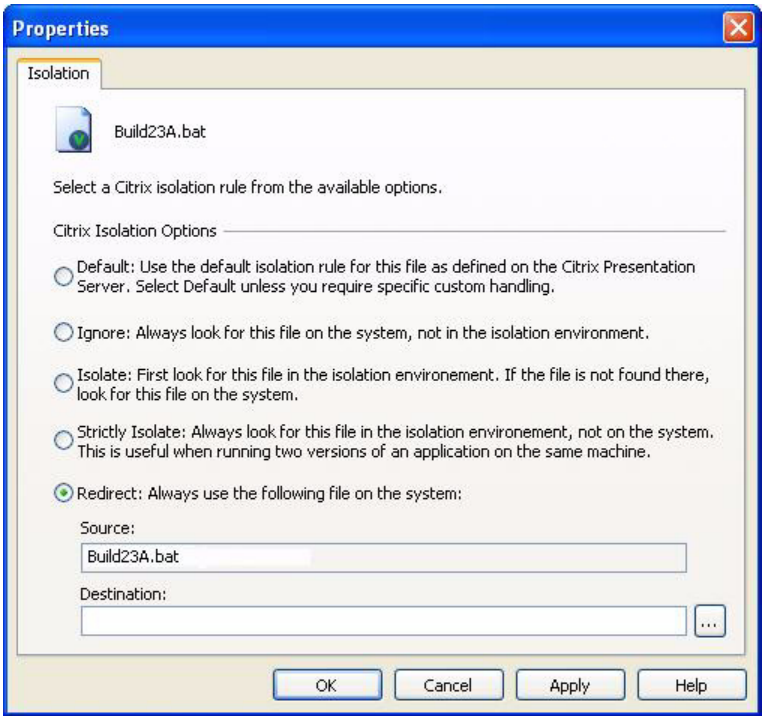


Figure 10-67: File Isolation Options Dialog Box



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.

The File Isolation Options dialog box includes the following options;

Table 10-59 • File Isolation Options



Option	Description
Default	Use the default isolation option for this file as defined on the Citrix XenApp.  Note • Select this option unless you require specific custom handling.
Ignore	Always look for this file on the system, not in the isolation environment.
Isolate	First look for this file in the isolation environment. If the file is not found there, look for this file on the system.

Table 10-59 • File Isolation Options

Option	Description
Strictly Isolate	<p>Always look for this file in the isolation environment, not on the system.</p>  <p>Note • This is useful when running two versions of an application on the same machine.</p>
Redirect	<p>Always use the following file on the system, not the one in the isolation environment. If you select this option, also specify the following:</p> <ul style="list-style-type: none"> • Source—Name of the selected file. • Destination—Select the file on the system that you want the application to use instead of the selected file.

Folder Isolation Options Dialog Box

On the **Folder Isolation Options** dialog box, you can override the default Citrix isolation option for the selected folder.

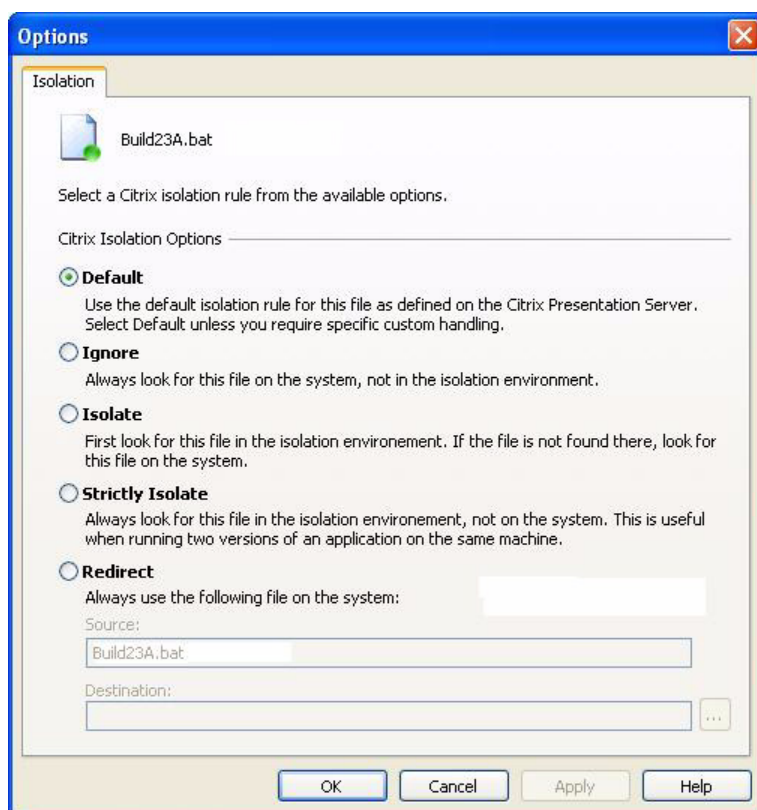




Figure 10-68: Folder Isolation Options Dialog Box



Caution • Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.

The **Folder Isolation Options** dialog box includes the following options;

Table 10-60 • Folder Isolation Options

Option	Description
Default	<p>Use the default isolation option for this file as defined on the Citrix XenApp.</p>  <p>Note • Select this option unless you require specific custom handling.</p>
Ignore	Always look for this file on the system, not in the isolation environment.
Isolate	Look for this folder in both the isolation environment and on the system. If the folder exists in both places, list both in the search results.
Strictly Isolate	<p>Always look for this folder in the isolation environment, not on the system.</p>  <p>Note • This is useful when running two versions of an application on the same machine.</p>
Redirect	<p>Always look in the following folder on the system, not in the one in the isolation environment. If you select this option, also specify the following:</p> <ul style="list-style-type: none"> • Source—Name of the selected folder. • Destination—The directory on the system where you want the application to look instead of looking in the selected folder in the isolation environment.

Registry Isolation Options Dialog Box

On the **Registry Isolation Options** dialog box, you can override the default Citrix isolation option for the selected registry key.

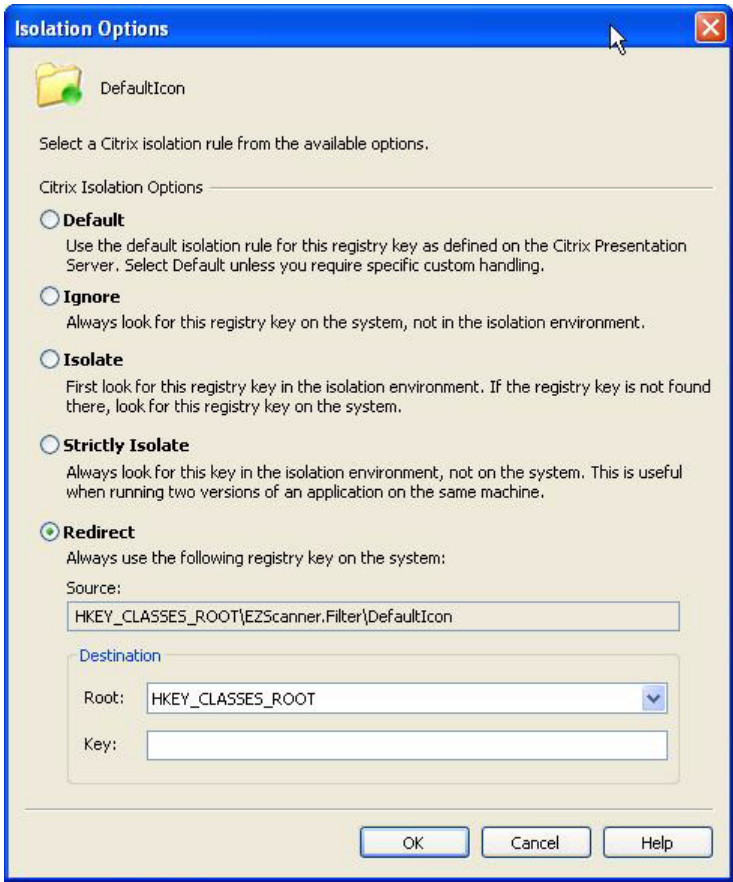


Figure 10-69: Registry Isolation Options Dialog Box



Caution • *Modify isolation options only if you have advanced knowledge of Microsoft operating system objects and registry settings.*

The Registry Isolation Options dialog box includes the following options;

Table 10-61 • Registry Isolation Options



Option	Description
Default	Use the default isolation option for this registry key as defined on the Citrix XenApp.  Note • <i>Select this option unless you require specific custom handling.</i>

Table 10-61 • Registry Isolation Options (cont.)

Option	Description
Ignore	Always look for this registry key on the system, not in the isolation environment.
Isolate	Look for this registry key in both the isolation environment and on the system. If the registry key exists in both places, list both in the search results.
Strictly Isolate	<p>Always look for this registry key in the isolation environment, not on the system.</p>  <p>Note • This is useful when running two versions of an application on the same machine.</p>
Redirect	<p>Always use the following registry key on the system. If you select this option, also specify the following:</p> <ul style="list-style-type: none"> • Source—Lists the name of the selected registry key. • Destination Root—Select the registry root of the registry key on the system that you want to redirect to. • Destination Key—Select the registry key on the system that you want to redirect to.

Service Packs Requirement Dialog Box

The **Service Packs Requirement** dialog box is opened by selecting an operating system on the **Profile Requirements** page of the Citrix Assistant and selecting **Service Packs Requirement** from the context menu.

On this dialog box, you can specify which, if any, Service Packs are required for the application to run on the selected operating system

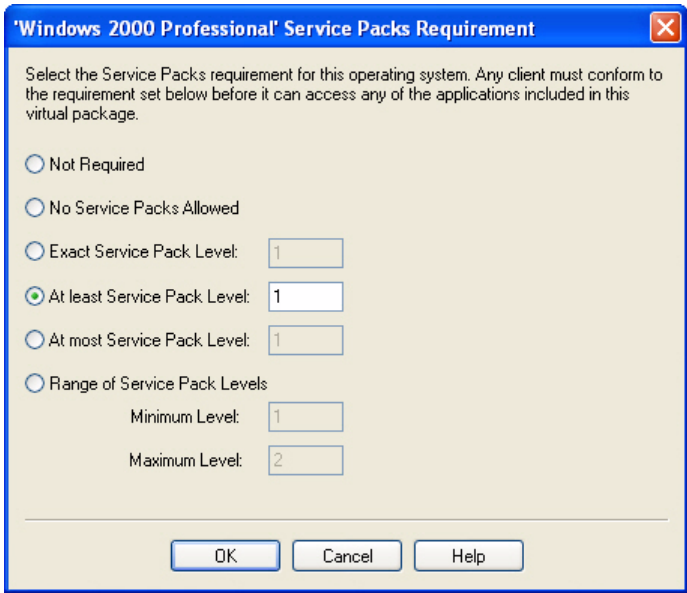


Figure 10-70: Service Packs Requirements Dialog Box

The **Service Packs Requirement** dialog box includes the following options:

Table 10-62 • Service Packs Requirement Options

Option	Description
No Service Pack Requirement	Select this option if this application supports all versions of this operating system, regardless of the number of Service Packs installed.
No Service Pack Allowed	Select this option if this application only supports the initial release of this operating system; if any Service Packs are installed, this application will not run properly.
Exact Service Pack Level	Select this option if this application requires the installation of a specific Service Pack on this operating system in order to run properly. Enter the required Service Pack Level in the box.
At Least Service Pack Level	Select this option if, to run properly, this application requires that this operating system have at least the specified Service Pack (or higher) installed. Enter the minimum required Service Pack Level in the box.

Table 10-62 • Service Packs Requirement Options

Option	Description
At Most Service Pack Level	Select this option if, to run properly, this application requires that this operating system have at most the specified Service Pack (or lower) installed. Enter the maximum required Service Pack Level in the box.
Range of Service Pack Levels	Select this option if, to run properly, this application requires that this operating system have a specified range of Service Packs installed. If you select this option, specify the Minimum Level and Maximum Level in the boxes.

Building Citrix Profiles Using the Command Line

When you configure a Citrix profile in an InstallShield project and then build that project (using either the user interface or the command line), both the Windows Installer package and the Citrix profile are built. When you use the standard InstallShield command line build, you do not need to add any additional command line parameters. All of the Citrix profile settings are saved within the InstallShield project.

Citrix Profile Conversion Error and Warning Messages



Note • For troubleshooting information about resolving errors and warnings that you may encounter when you are building a virtual application, see *Virtualization Conversion Errors and Warnings*.

Application Features Requiring Pre- or Post-Conversion Actions

Some application features are ignored when creating a Citrix profile. Therefore, some additional pre- or post-conversion actions must be taken in order for the application profile to run on Citrix XenApp.

One action you could take to try to include ignored features in a Citrix profile is to first repackage the application using the Repackaging Wizard, and then convert the repackaged application to a Citrix profile.

For a list of ignored features, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

For a list of ignored features, see [Application Features Requiring Pre- or Post-Conversion Actions](#).

Chapter 10: Creating Customized Virtual Applications

Creating Citrix Profiles

Customizing and Authoring Installations Using InstallShield



Edition • InstallShield Editor is included with AdminStudio Standard, Professional, and Enterprise Editions.

InstallShield Editor provides the most comprehensive and flexible setup-creation technology available for the Windows Installer. With the latest InstallShield installation development environment (Interface) you can create setup packages that utilize Windows Installer technology, while harnessing the flexibility provided by InstallScript, InstallShield's development language.

Administrators can also take advantage of InstallShield Editor to customize repackaged legacy setups, further enhancing them prior to deploying them in production environments. For AdminStudio Professional and Enterprise Editions, InstallShield Editor contains integrated Application Manager and ConflictSolver functionality, allowing you to perform conflict identification before leaving InstallShield Editor.

InstallShield Editor's documentation is divided into the following main areas:

Table 11-1 • InstallShield Editor Documentation

Section	Description
InstallShield Editor Integration with Application Manager and the Software Repository	Explains how to use the Application Manager submenu on the InstallShield Editor File menu to access packages stored in an AdminStudio Application Catalog Software Repository.
Application Manager Tab on InstallShield Editor Options Dialog Box	Describes the additional tab that appears on the InstallShield Editor Options dialog box when InstallShield Editor is invoked from ConflictSolver: the ConflictSolver tab.
Microsoft App-V, VMware ThinApp, and Citrix XenApp Virtualization Support	Provides an overview of how Repackager and InstallShield Editor provide support for the conversion of Windows Installer packages to virtual applications.

Table 11-1 • InstallShield Editor Documentation (cont.)

Section	Description
InstallShield Editor Help Library	Explains how to use InstallShield Editor to take advantage of its features to build Windows Installer packages.

AdminStudio-Specific Functionality in InstallShield Editor

When running InstallShield Editor from AdminStudio, some AdminStudio-specific functionality is enabled:

Table 11-2 • AdminStudio Functionality in InstallShield Editor

Functionality	Description
InstallShield Editor Integration with Application Manager and the Software Repository	<p>If you have the AdminStudio Enterprise Edition, which includes the Software Repository feature, you can perform the following tasks from the InstallShield Editor interface:</p> <ul style="list-style-type: none"> • Getting a Copy of a Software Repository Package • Opening and Editing a Software Repository Package • Adding a Package to the Software Repository via the InstallShield Editor Build Process • Using the Application Manager Context Menu
Repackager Integration	<p>You can launch Repackager from the InstallShield Editor interface for conversion of the following:</p> <ul style="list-style-type: none"> • Novell ZENworks projects (.axt/.aot) • Microsoft SMS Installer projects (.ipf) • WinINSTALL exported text files (.txt) • Wise Installation files (.wse) • Legacy Repackager projects (.inc) <p>When you attempt to open any of these file types in InstallShield Editor, you can launch Repackager to perform the conversion.</p>
<p>Application Manager options in the Options dialog box</p> <p><i>AdminStudio Professional and Enterprise Editions</i></p>	<p>The InstallShield Editor Options dialog box has an additional tab for Application Manager and ConflictSolver configuration.</p> <p>See Application Manager Tab on InstallShield Editor Options Dialog Box</p>

Table 11-2 • AdminStudio Functionality in InstallShield Editor (cont.)

Functionality	Description
Importing packages and merge modules into the AdminStudio Application Catalog from InstallShield Editor <i>AdminStudio Professional and Enterprise Editions</i>	Open the File menu, point to Application Manager , and click Add to open the Import Wizard . Using this Wizard, you can import Windows Installer packages and merge modules into the AdminStudio Application Catalog.
Checking for conflicts from InstallShield Editor <i>AdminStudio Professional and Enterprise Editions</i>	Open the File menu, point to Application Manager , and click Conflicts to open the Conflict Wizard . Using this Wizard, you can check for package conflicts before leaving InstallShield Editor.



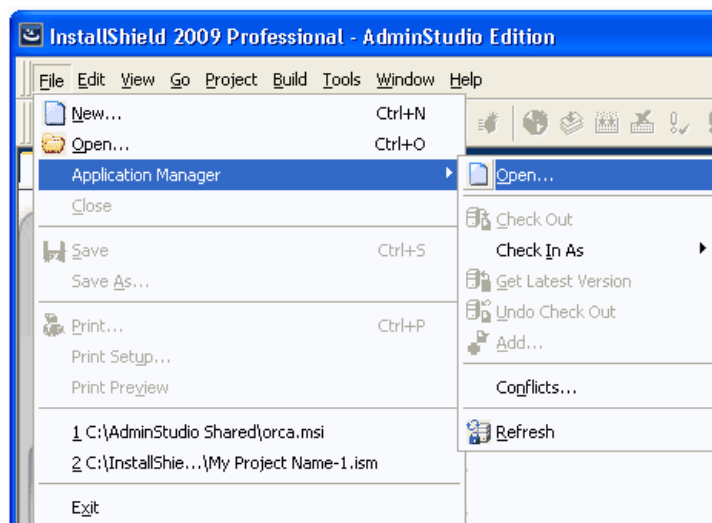
Note • AdminStudio Role permissions apply to Application Manager and ConflictSolver functionality in InstallShield Editor. If you are not assigned to a Role with sufficient permissions, you may not be able to access some of these features.

InstallShield Editor Integration with Application Manager and the Software Repository



Edition • The Software Repository feature is available in AdminStudio Enterprise Edition.

By making selections from the **Application Manager** submenu on the InstallShield Editor **File** menu, you can access packages stored in a AdminStudio Application Catalog Software Repository.



InstallShield Editor can perform the following tasks using packages in the Software Repository:

- [Getting a Copy of a Software Repository Package](#)
- [Opening and Editing a Software Repository Package](#)
- [Adding a Package to the Software Repository via the InstallShield Editor Build Process](#)
- [Using the Application Manager Context Menu](#)



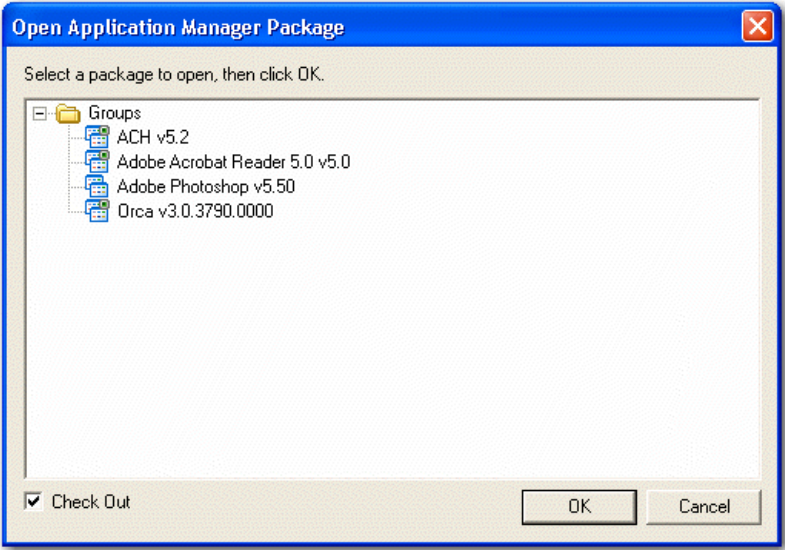
Note • For additional information, see [Using the Software Repository](#).

Getting a Copy of a Software Repository Package



Task: *To copy a package from the Software Repository and open it in InstallShield Editor:*

1. In AdminStudio, connect to an Application Catalog that has the Software Repository enabled.
2. Launch InstallShield Editor.
3. On the **File** menu, point to **Application Manager** and click **Open**. The **Open Application Manager Catalog** dialog box opens, listing the package tree in the AdminStudio Application Catalog:



In Application Manager, packages that are managed within the Software Repository have a different icon than those that are not:

Table 11-3 •

Icon	Description
	Package is not managed within the Software Repository.
	Package is managed within the Software Repository.
	Package is managed within the Software Repository and is checked out.
	Merge Module is managed within the Software Repository.

4. Clear the **Check Out** option.

5. Select a package that is stored in the Software Repository and click **OK**. A copy of the selected package is saved onto your local directory. When the copy is complete, the selected package is opened in **Direct MST Mode** and the **Installation Information View** is open.

You can now customize this file by creating a transform and save your changes onto your local directory. The original package in the Software Repository is not modified.

Opening and Editing a Software Repository Package

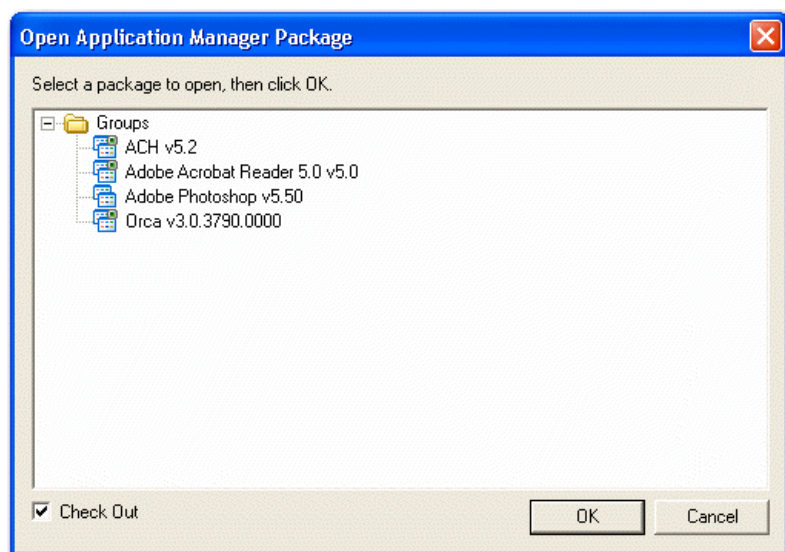


Edition • The Software Repository feature is available in AdminStudio Enterprise Edition.






Task: To open and edit Software Repository package in InstallShield Editor:

1. In AdminStudio, connect to an Application Catalog that has the Software Repository enabled.
2. Launch InstallShield Editor.
3. On the **File** menu, point to **Application Manager** and click **Open**. The **Open Application Manager Catalog** dialog box opens, listing the package tree in the AdminStudio Application Catalog:



In Application Manager, packages that are managed within the Software Repository have a different icon than those that are not:

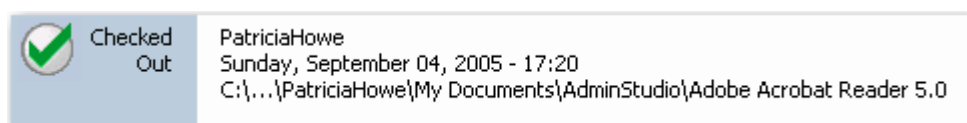
Icon	Description
	Package is not managed within the Software Repository.

Icon	Description
	Package is managed within the Software Repository.
	Package is managed within the Software Repository and is checked out.
	Merge Module is managed within the Software Repository.

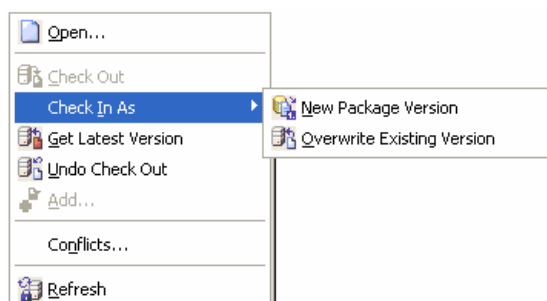
4. Select the **Check Out** option.
5. Select a package that is stored in the Software Repository and click **OK**. A copy of the selected package is saved onto your local directory.

When the copy is complete, the selected package is opened in **Direct MST Mode** and the **Installation Information View** is open.

The package is now in Checked Out status in Application Manager, meaning that no one else can check it out until you check it back in. If you were to open Application Manager and select this package, you would see the Checked Out icon in the Product View:



6. Customize this package using InstallShield Editor.
7. When you have finished editing this package, on the File menu, point to Application Manager and click **Check in As**. You are now given two options: **New Package Version** or **Overwrite Existing Version**.



8. Select one of those options to save the edited package in the Application Catalog.

Adding a Package to the Software Repository via the InstallShield Editor Build Process



Task: *To add a package to the Software Repository via the InstallShield Editor Build process:*

1. In AdminStudio, connect to an Application Catalog that has the Software Repository enabled.
2. Launch InstallShield Editor.
3. Open an InstallShield project (.ism).
4. Open the **Releases View**.
5. Set the **Publish to Software Repository** property to **Yes**.
6. Select the **AdminStudio Group** that you want the imported package to belong to.
1. Select one of the following **Duplicate Package** options:
 - **New Package Version**
 - **Overwrite Existing Version**
 - **New Package History Version**
 - **Ignore if Exists**

2. Build the setup.

Upon completion of the build, the MSI package is published to the Software Repository. The progress messages are displayed in the Output window.



Note • *If the setup is compressed, an administrative image must be created before the package can be published to the Software Repository.*

Using the Application Manager Context Menu

Instead of opening packages from the **Open Application Manager Catalog** dialog box in InstallShield Editor, you can select a Software Repository package in the Application Manager tree and select **Edit with InstallShield Editor** from the context menu. The package is checked out to you and opened in the Direct MST Mode of InstallShield Editor.

Application Manager Tab on InstallShield Editor Options Dialog Box

The InstallShield Editor **Options** dialog box has a tab named **Application Manager**.

From this tab, you can specify the Application Catalog database to which you want to connect when using integrated Application Manager functionality in InstallShield Editor. If you click the folder icon, the **Connect Application Catalog** dialog box opens, from which you can select the Application Catalog.

If you click **Advanced**, the [Options Dialog Box](#) opens.

Microsoft App-V, VMware ThinApp, and Citrix XenApp Virtualization Support



Edition • The Windows Installer Converter, Microsoft App-V Assistant, ThinApp Assistant, and Citrix Assistant are included in the AdminStudio Virtualization Pack.



Important • AdminStudio ThinApp support requires a separate purchase of VMware ThinApp™.

Both Repackager and InstallShield Editor provide support for the conversion of Windows Installer packages to virtual packages:

- **Windows Installer Converter**—You can use the Windows Installer Converter in Repackager to convert a Windows Installer package to a Microsoft App-V application, ThinApp application, or Citrix profile. You would use this tool if you wanted to convert a Windows Installer package to a virtual package with no file modifications and using the default isolation options. However, if you decide that you want to customize the virtual package, you can click a button in the Windows Installer Converter to open that package in the InstallShield Microsoft App-V Assistant, ThinApp Assistant, or Citrix Assistant.
- **Repackager Interface**—By selecting an option on the Repackaged Output view, you can simultaneously build an InstallShield Editor project, a Windows Installer package, a Microsoft App-V application, a ThinApp application, and a Citrix profile from a Repackager project.
- **Microsoft App-V Assistant**—Using the Microsoft App-V Assistant, you can convert a Windows Installer package or an InstallShield project to a customized App-V application. You can modify a Microsoft App-V application's operating system requirements, files, folders, shortcuts, registry settings, isolation options, and build options.
- **ThinApp Assistant**—Using the VMware ThinApp Assistant, you can convert a Windows Installer package or an InstallShield project to a customized ThinApp application. You can configure a ThinApp application's files, folders, shortcuts, registry settings, isolation options, and build options.

- **Citrix Assistant**—Using the Citrix Assistant, you can convert a Windows Installer package or an InstallShield project to a customized Citrix profile. You can modify a Citrix profile's operating system and language requirements, files, folders, shortcuts, registry settings, script execution, isolation options, and build options.

For more information on the capabilities of these features, see [Getting Started With Application Virtualization](#).

Differences Between InstallShield Editor and InstallShield 2009 Professional Edition

The InstallShield Editor that is included with AdminStudio 9.5 is based upon InstallShield 2009 Professional Edition, but it has a slightly different feature set. Those differences are explained here.

Default Project Types

In InstallShield Editor, all non-Windows Installer-based project types are disabled by default. You can enable these additional project types, such as InstallScript, on the InstallShield Editor **Options** dialog box, which is opened by clicking **Options** on the **Tools** menu.

Multilingual Runtime Language Support

InstallShield Editor includes InstallShield 2009 Premier Edition's multilingual runtime language support, which enables you to create a single installation that displays end-user text in multiple languages. If an installation will contain more than one language, you can specify whether to prompt the end user to select the run-time language, or to automatically display the language of the target system's operating system.

Using a Network Repository to Share Project Elements

A repository is a collection of common elements that can be shared and reused in different installation projects, enabling you to ensure consistency. InstallShield Editor includes not only InstallShield 2009 Professional Edition's *local* repository support, but also InstallShield 2009 Premier Edition's *network* repository support, which fosters collaboration among installation authors.

InstallShield Editor Help Library



Edition • *InstallShield Editor is included with AdminStudio Standard, Professional, and Enterprise Editions.*

The InstallShield Editor Help Library gives you unified access to InstallShield Editor Help. This library's help topics contain information that assist you in finding answers with InstallShield Editor.

Open the [InstallShield Editor Help Library](#) to see a listing of the Help Library's contents.



Note • *You can also download the InstallShield documentation PDFs from the [InstallShield Documentation Center](#).*

Customizing Installations Using Tuner

Using Tuner, you can add to, modify, or remove information from a Windows Installer package. This involves creating a transform file, where all the modifications are stored. When you install the package and transform together, your modifications are reflected in the installation.

Tuner allows you to configure the initial state of features, add or remove files from an installation, edit registry entries, configure setup properties, set Add/Remove Programs options, and configure servers for application resiliency. You can also validate Windows Installer packages and transform files to ensure they conform to Microsoft guidelines.

Tuner user documentation is presented in the following sections:

Table 12-1 • Tuner User Documentation

Section	Description
Working with Transforms	Explains how to create a transform file to customize a Windows Installer-based installation.
Validation	Explains how to compare a Windows Installer-based installation to a known set of guidelines (an evaluation file) to ensure it has been created to those guidelines.
Setup Organization	Explains how to modify two main parts of the installation that your end users will see: the default path and default company name, and the actual features that can, will, or will not be installed.
Configuring Package Content	Explains how to use a transform file to manipulate the original package contents, including files and folders, registry entries, shortcuts, INI files, ODBC resources, and NT services.
Working with Dialogs	Explains how to disable particular panels that appear during the installation, administrative, patch, or maintenance sequences.

Table 12-1 • Tuner User Documentation (cont.)

Section	Description
Configuring Additional Server Locations	Explains how to configure additional server locations.
Changing Add/Remove Program Settings	Explains how to configure the Windows setup to give the user the option of removing, repairing, or changing the installation with the click of a button.
Customizing Setup Properties	Explains how to edit property values in the properties table (the underlying structure of Windows Installer packages). You can also add your own custom properties.
Preparing Packages for Distribution	Explains how to postvalidate your transform and base Windows Installer package, and how to package the transform and base package for distribution.
Directly Editing Packages	Explains how to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform.
Tuner Reference	Information on Tuner Views and dialog boxes.

When to Use Tuner vs. InstallShield Editor

Most system administrators use InstallShield Editor to import repackaged setups and convert them into Windows Installer packages. InstallShield Editor is also ideal for making changes to the package that you want reflected in all deployments of the package. However, it is recommended that you use Tuner to create transforms for changes that you only want to affect a particular deployment, rather than every installation.

Working with Transforms

The Microsoft-designated term transform refers to a specific file type used to customize a Windows Installer-based installation. A transform contains all modification information, such as whether features are installed, how they are installed, which files, shortcuts, and registry entries are included, and Windows 2000 and XP Add/Remove Programs information. Transform files use an .mst extension.

For example, you may need to customize an installation for different departments in your company. Typical business productivity suites come with a spreadsheet program, a word processor, and a presentation tool. Your accounting department may only need the spreadsheet and the presentation programs. On the other hand, the writing department may need only the word processor and the spreadsheet. A third department may need the entire suite of applications. Instead of manually setting up every person in the company, you can take the original setup of the entire suite, and create a customization project in the form of a transform to meet the needs of each department. A transform would need to be created for every configuration that you plan to use.

Once you have created a transform, you can apply it at runtime, depending on whose machine the application is being installed on. For example, in the accounting department, the transform limits the installation to include only the spreadsheet and presentation programs.

In some cases, it may be necessary to have multiple transform files for an installation. For example, a vendor may use a transform file for language-specific information. When you want to customize that MSI file, you need to include the preexisting transform so your modifications affect the entire existing package.

Creating New Transform Files



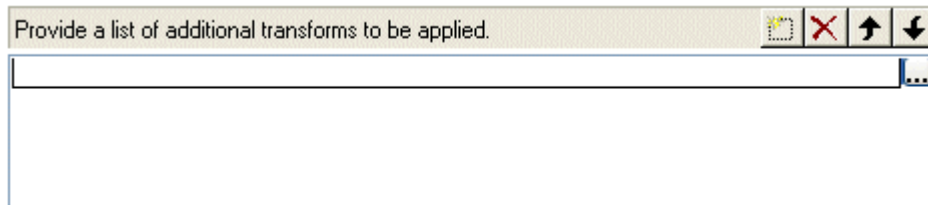
Task:

To create a new transform file:

1. Launch **Tuner** from the AdminStudio interface. The **Tuner Start Page** opens.
2. Click **Create a new transform** on the left side of the view or select **New** from the **File** menu. The right side of the view changes to display the fields necessary to create a new transform.
3. In the **Select an MSI File** field of the **Base Windows Installer Package** area, enter the name and location of the Windows Installer package that you are customizing, or click **Browse** to locate it.
4. If there are transforms already associated with the Windows Installer package, (for example, previous customizations or transforms containing language-specific information), go to the **Provide a list of additional transforms to be applied** area and click the New button:



A new entry appears in the list.



When an entry appears in the list, click the Browse button (...) to the right of it and locate the transform. If multiple transforms are associated with this package, use the Move Up and Move Down buttons to specify the order in which the transforms are applied.



Caution • When using multiple transforms, keep in mind that the order in which they are applied is critical. For example, if you create a transform for a Windows Installer package that creates a new value for a property, and then create a second transform that changes the value created in the first transform, everything works correctly. However, if you apply the second transform first, that transform is attempting to modify the property's value, instead of creating it. That will result in an error.

One simple example of where this may be a problem is with the default company name. If the value is not set by default, and you set it in using the first transform, a new value for the property is created. If you create a second transform that modifies the combined original package and first transform, and the second transform changes the default company name, it is only changing the property. However, if you try to apply the second transform

without the first one, Windows Installer interprets this as trying to change a null value to another value, which will result in an error.

5. By default, the transform will be created in the same directory as the Windows Installer package, and named the same as the base package with an .mst extension. However, if you want to change the name and/or location of the transform, you can also do so in the **Windows Installer Transforms** area. Click **Browse** to open the **Save Customization File** dialog box.

Navigate to the directory in which you want to store the transform file you are creating. Provide the name of the transform with an .mst extension (for example, MyTransform.mst) and click **Save**. The dialog box closes and the path and file name appear in the edit field.

6. If you want to create a **Response Transform**, check the appropriate box. If you are using a response transform, you can specify additional command-line properties (in property name/value pairs separated by semicolons) to pass to the response transform. These must be PUBLIC properties, and only control how the dialogs are displayed during creation of the response transform. They are not persisted outside of the UI sequence during creation. For example, you can pass the property/value pair ARPHELPTTELEPHONE=1-111-111-1111 to set the value of the **Help Telephone** field of **Add/Remove Programs**. See [Using Response Transforms](#) for more information.

You might pass a property/value pair during response transform creation to display all dialogs during an installation that may not be displayed based on your system configuration (for example, to show Windows 9x-only dialogs on a Windows NT platform). You can then make appropriate responses and have them included in your transform.

7. Ensure all the information entered is correct, and click **Create**.
 - **If you are creating a Standard Transform**, the transform file is opened in the Tuner interface, displaying the [Package Validation View](#).
 - **If you are creating a Response Transform**, a simulated installation of the selected application begins. Step through the installation, making changes as necessary. When you reach the end of the installation sequence and click Install, the installation will exit and the Tuner interface will open your transform, displaying the [Package Validation View](#). Your transform contains all of the changes you made during the simulated installation.



Note • You can access information about the original MSI file and associated transforms by selecting **Properties** from the **File** menu.

Opening Existing Transforms

On the Tuner Start Page, when you click on Open an existing transform file, the pane on the right of the interface changes. You can then specify the name and location of the base Windows Installer package, any associated transforms, and the name and location of the transform file.



Note • Generally, you will only use this option when opening existing transforms that were created by a product other than Tuner, or created by someone other than yourself. Transforms you create using Tuner are more easily accessed through using the Open a recent transform file selection.

Opening Recently Accessed Transforms

On the Tuner Start Page, when you click on Open a recent transform file, the pane to the right changes to a list containing your most recently accessed transforms.

From this View, you can perform the following tasks:

- To open a transform file, select a transform file and click Open.
- To view information on the transform file, select a transform file and click Properties. The Properties dialog box opens, listing details about the base MSI package and associated transforms.
- To specify the view that will appear when Tuner is started, select one of the following options:
 - Reload the last project saved when restarting Tuner
 - Make this my default Tuner Start Page Screen
 - Make Welcome my default Tuner Start Page Screen

Creating Generic Transforms

Most transforms are tied to a specific product code, meaning they can only be applied to a specific version of a product. Generic transforms, however, do not have that limitation. They can be created to apply to multiple versions of a Windows Installer package (for example, Office XP and Office 2000), or to any Windows Installer package.



Task:

To create a generic transform:

1. Create a transform in Tuner as you would do normally.
2. With the transform project open, select Transform Summary Information from the Project menu to access the **Transform Summary** dialog box.
3. When the **Transform Summary** dialog box appears, change the validation options to reflect how you want the transform applied. If you want to create a completely generic transform, deselect all validation options.



Tip • One use for generic transforms is to enforce standard Add/Remove Programs information on every package installed in your environment. The same transform can be used to set all relevant properties.

Using Response Transforms

There are two ways you can create a transform file in Tuner:

- The first and the most common way is to begin by creating an empty transform and then making customization by navigating to different views in Tuner.
- The second way is by running the installation and then customizing various options available in each setup panel. The installation is only simulated and no changes take place on the user system. Tuner saves all the changes that the user has made on each panel of the setup in the transform. This type of transform is called a response transform.

Response transforms, much like installation response files, allow you to run an existing Windows Installer-based installation and capture your configurations. Unlike a response file, these changes are used as a starting point for your new transform.

For example, if you use Tuner to create a response transform, you might select certain features you want installed, the location of the installation, and company information. When the Tuner interface is opened, these values will already be set for your new transform file. You can then make further customizations as necessary.

You might want to create a response transform for an installation, and then fill in your company name as the default name, and a specific directory for installation which is different from the one suggested by the manufacturer. Further, you may want to configure a specific feature, such as clip art, to not be installed. By using the familiar installation user interface, you can quickly make your basic customizations before using the Tuner environment to refine the transform.

Viewing Transform Properties

To view properties of the transform you are currently creating or editing, select Properties from the File menu. To view the properties of a project from the [Open a Recent Transform View](#), select the transform file and click Properties or selecting Properties from the context menu.

The resulting Properties dialog box provides information about the transform, including the name and location of the base Windows Installer file, and any additional transforms that are associated with this transform and MSI file.

Validation

What is Validation?

Validation is the process of comparing a Windows Installer-based installation to a known set of guidelines (an evaluation file) to ensure it has been created to those guidelines. Tuner can perform two types of validations: prevalidation and postvalidation.

- **Prevalidation** compares only the base Windows Installer package to an evaluation file. This ensures that, when starting a customization project, the initial file was created using the guidelines in that evaluation file. If it does not pass prevalidation, then the installation *may* work fine, but it may not be able to use all Windows Installer features.

- **Postvalidation** compares the base Windows Installer package and the changes made in a customization project against an evaluation file. In this case, the combination of the initial file and the subsequent modifications of the transform can produce different results than a comparison of just the base Windows Installer package. If the initial file was valid and postvalidation fails, the problems exist in the customization project. In some situations, advanced users may be able to use a transform file to make an initially invalid MSI file valid in conjunction with the transform.

What Do You Validate Against?

Tuner provides two files that you can validate against: the Windows 2000 Logo Program Suite and the Full MSI Validation Suite.

- The Windows 2000 Logo Program Suite is a subset of the Full MSI Validation Suite, and is used to certify that the installation meets the Microsoft standards for the Windows logo.
- The Full MSI Validation Suite is used to ensure that the installation meets all MSI standards.

Validation Procedure

Follow this procedure when validating an installation:

- In practice, you should prevalidate the base MSI file to ensure compliance to MSI standards before you begin creating a customization project.
- After you have finished your project, postvalidate it to make sure it is still compliant.
- If the base installation was compliant and the postvalidation fails, go back to the changes you made to determine what caused the validation problems.

For full details, consult the [Windows Installer Help](#).

Prevalidating Windows Installer Packages

Prevalidation compares only the base Windows Installer package to an evaluation file. This ensures that, when starting a customization project, the initial file was created using the guidelines in that evaluation file.

Performing a Prevalidation



Task: *To prevalidate a Windows Installer package:*

1. After creating a new transform file and specifying the base Windows Installer package, select the Prevalidation view from the checklist. The **Prevalidation View** appears, listing the name of the base Windows Installer Package.
2. Specify or browse to the Evaluation File you want to use.
3. If you want to run specific [Internal Consistency Evaluators](#) (ICEs), specify them in the ICEs to Run text box, separating them by semicolons if there are more than one (for example, ICE07;ICE13;ICE72). Otherwise, all ICEs are used.

4. Specify the result level by checking the Show “INFO” messages, Show “WARNING” messages, and/or Show “ERROR” messages check boxes. It is highly recommended that you check at least the Error check box so you are certain you are not suppressing results that occur in invalid packages.
5. Click the MSI Validation button (🔍) on the toolbar, or click the Start button in the view.

Viewing the Prevalidation Results

As each ICE is run, Errors, Warnings, and Info messages are generated, and are listed in the Output tab at the bottom of the interface.

Upon completion of the Prevalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (❌), a Warning (⚠️), or an Informational Message (ℹ️), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row’s associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red.

This feature makes it very easy for you to use the **Direct Editor** to edit values in the MSI tables of the base Windows Installer package and store them in your transform. For more information, see [Directly Editing Packages](#).



Note • If no errors appear in the results (providing you are displaying errors), then the package is valid against the specific ICEs you specified, or against the entire evaluation file (if no ICEs were selected).

Handling Invalid Windows Installer Packages

Ideally, all Windows Installer packages will pass validation. Realistically, many will fail (generating errors). When a package fails validation, it means the package was not built to Microsoft’s specifications. It does not mean the installation does not work. However, there are a few things you can do when your package has validation errors:

Table 12-2 • Methods to Resolve Validation Errors

Solution	Explanation
Use Tuner to correct validation errors.	This involves opening the base package using Tuner and creating a transform file which contains your corrections.
Contact the installation vendor.	The company that created the installation (usually the same company that created the software) may be able to resolve the validation issues and provide you with a valid setup. Be sure to provide the validation report to vendors so they know where to focus.
Reconsider using the application.	Although it might be an extreme reaction to an invalid package, there may be compelling reasons not to use an installation not built to Microsoft guidelines.

Table 12-2 • Methods to Resolve Validation Errors (cont.)

Solution	Explanation
Ignore the problems and install anyway.	This is probably the most likely scenario. The invalid installation may not be worth trying to fix, or even have errors that you are concerned about. You could proceed and just use the installation as it is. From a practical standpoint, this may be your best option.




Note • Most packages will also generate Warnings during validation. These can occur in valid packages, and many cannot be removed. Although the presence of Warnings does not make a package invalid, it is generally a good practice to eliminate Warnings (if possible).

Postvalidating Transforms






Task:

To postvalidate a Windows Installer package and the transform you are creating:

1. Select the Postvalidation view from the checklist. The Postvalidation view appears, listing the name of the base Windows Installer Package.
2. Specify or browse to the Evaluation File you want to use.
3. If you want to run specific [Internal Consistency Evaluators](#) (ICEs), specify them in the ICEs to Run text box, separating them by semicolons if there are more than one (for example, ICE07;ICE13;ICE72). Otherwise, all ICEs are used.
4. Specify the result level by checking the Show “INFO” messages, Show “WARNING” messages, and/or Show “ERROR” messages check boxes. It is highly recommended that you check at least the Error check box so you are certain you are not suppressing results that occur in invalid packages.
5. Click the Transform Validation button () on the toolbar, or click the Start button in the view.

Viewing the Postvalidation Results

As each ICE is run, Errors, Warnings, and Info messages are generated, and are listed in the Output tab at the bottom of the interface.

Upon completion of the Postvalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error () , a Warning () , or an Informational Message () , the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row’s associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red.

This feature makes it very easy for you to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform. For more information, see [Directly Editing Packages](#).



Note • If no errors appear in the results (providing you are displaying errors), then the package and transform are valid against the specific ICEs you specified, or against the entire evaluation file (if no ICEs were selected).



Tip • It is possible for a package that passed the prevalidation to fail the postvalidation. Remember changes made in the Setup Properties can affect your installation. If your package fails postvalidation, check all changes made in the Setup Properties for accuracy. To identify the original Setup Properties, you can create a new transform file that can be deleted at any time. Changes made using the Direct Editor can also affect your installation's functionality.

Evaluation Files and Internal Consistency Evaluators

When you prevalidate the base Windows Installer package, or postvalidate the package and your transform, Tuner runs several [Internal Consistency Evaluators](#) (ICEs) contained in the specified evaluation file. If the base package or your transform and package does not pass one of these ICEs, Tuner reports the failure. If the problem is in the base package, you can contact the software vendor to report the problem.

Setup Organization

The [Organization View](#) allows you to modify two main parts of the installation that your end users will see: the default path and default company name, and the actual features that can, will, or will not be installed.

Topics in this section include the following:

- [Changing a Feature's Visibility](#)
- [Setting the Initial State of a Feature](#)
- [Editing a Feature's Description](#)
- [Setting the Default Destination](#)
- [Setting the Default Organization](#)
- [Changing the Destination Variable](#)
- [Preventing Features from Displaying During Custom Installation](#)
- [Setting Feature Properties](#)
- [Using Feature Advertisement](#)

Changing a Feature's Visibility



Task: *To change the visibility of a feature:*

1. Under Organization in the checklist, select the Features View. This project's Features are listed in the second column.
2. Select the feature that you would like to change the visibility on. The Properties for that feature are listed.
3. Click in the Visible property in the Feature Properties grid and make a selection from the drop-down menu. Your options are:

Option	Description
Not Visible	The feature will not show up in the custom setup dialog box during installation.
Visible and Expanded	The feature will be displayed with all its subfeatures visible in the custom setup dialog box during installation.
Visible and Collapsed	The feature will be displayed in a collapsed state in the custom setup dialog box during installation.

Setting the Initial State of a Feature



Task: *To set the Initial State of a feature:*

1. Under Organization in the checklist, select the Features view. This project's Features are listed in the second column.
2. Select the feature that you would like to change the initial state of. The Properties for that feature are listed.
3. Click in the Initial State property in the Feature Properties grid and choose one of the selections from the drop-down menu. Your selections are:
 - The feature is not installed: By default, the feature will not be installed during setup.
 - The feature is installed on the local drive: By default, the feature will be installed on the local drive during setup.
 - The feature is run from source, CD, or the network: By default, the feature will be run from the source, whether it be from the installation CD or from the network.
 - The feature is advertised: By default, the feature will be advertised, but not installed. Essentially, this is an on-demand option; a shortcut will be created during setup, and if the shortcut is clicked, the feature will then be installed from the source. This ensures features that may be unnecessary are not installed until they are needed, if ever. For more information, see [Using Feature Advertisement](#).



Note • The initial default settings run the Setup in a quiet mode.

Editing a Feature's Description



Task: **To edit a feature's description:**

1. Under Organization in the checklist, select the Features view. This project's Features are listed in the second column.
2. Select the feature that you would like to change the description of. The Properties for that feature are listed.
3. Click in the Description property in the Feature Properties grid.
4. Enter the new feature description in the Description value cell.

Setting the Default Destination



Task: **To specify the Default Destination Path for an installation:**

1. Under Organization, select the Product Properties view from the checklist. The Product Properties view appears.
2. Click in the Default Destination Path property in the Product Properties grid.
3. Provide the path that you want to use as the Default Destination Path.



Caution • Consult the [Product Properties View](#) help topic for important information about the Default Destination Variable and how it can be affected by changing this value.

Setting the Default Organization



Task: **To specify the default organization for the installation:**

1. Under Organization, select the Product Properties view from the checklist. The Product Properties View appears.
2. Click in the Company Name property in the Product Properties grid.

3. Enter the name you want to use as the default organization name. The organization name can be a maximum of 30 characters in length.

Changing the Destination Variable



Task: *To specify the Destination Variable that holds the Default Destination Path:*

1. Under Organization, select the Product Properties view from the checklist. The Product Properties view appears.
2. Click in the Default Destination Variable property in the Product Properties grid.
3. Select the Default Destination Variable you want to use from the drop-down menu.



Caution • Consult the [Product Properties View](#) help topic for important information about this variable.

Preventing Features from Displaying During Custom Installation



Task: *To prevent a feature from being displayed to your end users during a custom installation:*

1. Under Organization in the checklist, select the Features View. This project's Features are listed in the second column.
2. Select the feature that you would like to hide. The Properties grid for that feature appears.
3. Click in the Visible property in the Properties grid and select Not Visible from the drop-down menu.

The feature selected will not be visible during custom installation. Depending on the feature's Initial State, the feature may or may not be installed on the end user's system.

Setting Feature Properties



Task: *To set Feature Properties:*

1. Select Features under Organization in the checklist.
2. Select the Feature that you want to edit. The Feature Properties view appears.
3. In the Description text box, enter a description that will be displayed when a feature is clicked in the Custom Setup dialog box

4. From the Visible drop down list, select an option to specify how the feature is presented to the end user in the Custom Setup dialog. The following options are available:

- **Visible and Collapsed:** The feature will be displayed in the Custom Setup dialog with its subfeatures collapsed by default.
- **Visible and Expanded:** The feature will be displayed in the Custom Setup dialog with its subfeatures expanded by default.
- **Not Visible:** The feature will not be displayed to the end user in the Custom Setup dialog.

Although an end user obviously cannot select or deselect an invisible feature, this property does not have any direct bearing on whether a feature is installed. In other words, a feature is not automatically installed if it is invisible; it just cannot be deselected if it would otherwise be installed, or selected if it should not be installed.

5. From the Initial State list, select an option to determine how (or if) the feature is installed during installation:

- **The feature is not installed (INSTALLSTATE_ABSENT):** The feature will not be installed during setup.
- **The feature is installed on the local drive (INSTALLSTATE_LOCAL):** The feature will be installed on the local drive during setup.
- **The feature is run from source, CD, or network (INSTALLSTATE_SOURCE):** The feature will be run from the source, whether it is from the installation CD or from the network.
- **The feature is advertised (INSTALLSTATE_ADVERTISED):** The feature will be advertised, but not installed. Essentially, this is an on-demand option; a shortcut will be created during setup, and if the shortcut is clicked, the feature will then be installed from the source. This ensures features that may be unnecessary are not installed until they are needed, if ever. For more information, see [Using Feature Advertisement](#).

Using Feature Advertisement

Windows Installer supports many features of Windows 2000 and later platforms, including feature advertisement. This convenience enables any product feature to be in one of four installation states:

- The feature is not installed
- The feature is installed on the local drive
- The feature is run from source, CD, or the network
- The feature is advertised

When features are advertised, they are not actually installed on the local system. However, they appear to be, in that the appropriate shortcuts to launch the feature are present. The first time a user attempts to use a feature that is advertised, it is installed on the computer.

Feature advertisement is fully supported only on the Windows XP, Windows 2000, and Windows Me operating systems, though an updated shell is available for Windows 95, Windows 98, and Windows NT 4.0. To activate the shell for any of these platforms, Microsoft's Internet Explorer 4.01 Service Pack 1 must be installed and Active Desktop enabled.



Note • For instructions on how to specify feature advertisement and the other installation states of a feature, see [Setting the Initial State of a Feature](#).

Configuring Package Content

Many modifications you can make in a transform file involve manipulating the original package contents. This includes the following:

- [Files and Folders](#)
- [Registry Entries](#)
- [Shortcuts](#)
- [INI Files](#)
- [ODBC Resources](#)
- [NT Services](#)

Files and Folders

From the Files and Folders view, you can perform all file operations in Tuner. This includes viewing files in the source MSI package, adding new files, preventing files from being installed, and removing added files.

Topics in this section include the following:

- [Adding Files](#)
- [Displaying Files from the Base Windows Installer Package](#)
- [Preventing Installation of Files from the MSI](#)
- [Removing Added Files](#)
- [Storing Added Files](#)

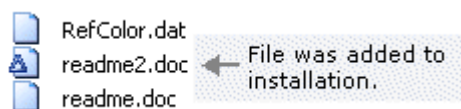
Adding Files



Task: *To add files to an installation:*

1. Select the Files and Folders view from the checklist. The Files and Folders View appears.
2. Navigate to the location in the Source computer's directory tree that contains the file you want to add.
3. Select the file you want to add from the Source computer's files pane.

4. Drag the file to the appropriate folder in the Destination computer's folders tree. The file then appears in the Destination computer's files pane, with an icon indicating that it is an added file.



Displaying Files from the Base Windows Installer Package



Task: *To display files from the base Windows Installer package in the Files and Folders view:*

1. Select Options from the Tools menu to display the Options dialog box.
2. Select the View Settings tab. The Option Dialog's View Settings pane opens.
3. Select the Display files from the original MSI package in addition to files added in the transform check box.
4. Click OK.

When you return to the Files and Folders view, all files from the base Windows Installer package are displayed as well as the files added in the transform. By default, this option is enabled.

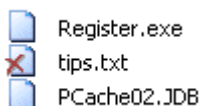
Preventing Installation of Files from the MSI



Task: *To prevent files from the base Windows Installer package from being installed:*

1. Select Files and Folders from the checklist. The Files and Folders View opens.
2. Navigate to the folder which contains the file in the Destination computer's folders tree that you want to remove from the installation.
3. In the Destination computer's files pane, select the file you want to prevent from being installed and select Remove from the context menu.

The actual file within the Windows Installer package is not deleted—only the entry in the File table is removed. The icon for the file changes to a computer with a red "X" over it.



If you remove a file, and later want to restore it to the installation, simply right-click the file again and select Restore.



Note • Key files in the base Windows Installer package are denoted with a key icon, and cannot be marked for deletion. Additionally, files contained within cabinet (.CAB) files are not displayed, and therefore cannot be marked for deletion.

Removing Added Files



Task: *To remove files you have added to an installation:*

1. Select Files and Folders from the checklist. The Files and Folders View opens.
2. In the Destination computer's folders tree, navigate to the folder containing the added file that you want to remove.
3. Select the file you want to remove from the Destination computer's files pane.
4. Select Remove from the context menu or press the Delete key.

Storing Added Files

When you add files to a transform, Tuner stores them in a CAB file with the same name as your transform. The added files are placed in the CAB and Tuner no longer maintains a reference to the original file location on the source computer. If you add additional files after saving the transform, the contents of the CAB file are extracted and recompressed along with the new files.

Because this mechanism relies on the presence of the CAB file, this file must be stored in the same location as the transform. If you move, modify, or delete the CAB file, Tuner will no longer be able to include the added files in the transform. You must then delete the files from the Files and Folders View and re-add them from their original locations, or locate the original CAB and place it back in the same folder as the transform.

Also, because the contents of the CAB file are uncompressed and recompressed when you add subsequent files to the installation, you must have sufficient disk space for this extraction when you save the transform.

Registry Entries

You can use the Registry view to create keys and values similar to how you use the Windows Registry Editor, or you can copy or drag and drop existing keys and values from the Source view.

Further, you can use this view to import an existing REG file using the Registry Import Wizard. You can also modify or delete registry keys that are part of the base installation. If you add new registry keys, they will always be installed.

Topics in this section include the following:

- [Creating a Registry Key](#)
- [Creating a Registry Value](#)
- [Importing REG Files](#)
- [Removing Registry Information](#)

Creating a Registry Key



Task: *To specify a registry key that will be created on the target system:*

1. Select Registry from the checklist. The Registry View opens.
2. In the Destination Computer Registry View, select the key to which you want to add a value. All existing values for that key in the Destination Computer Registry Data pane are displayed.
3. To create a subkey, select a registry hive (such as HKEY_CURRENT_USER) or an existing key, point to New on the context menu, and select Key.

A new key is created with the name “NewKey n ” (where n is a successive number).

4. Enter a meaningful name now to rename the key. If you want to change the name later, right-click on the key and select Rename.
5. Right-click on the new key and select whether the key is to be created on installation, deleted on uninstallation, or both.

Choose one of the following:

Option	Description
Create Key at Install	Creates the new registry key during installation if the key does not exist on the target machine.
Delete Key at Uninstall	Deletes the registry key during uninstallation, regardless of whether they key existed prior to the MSI's installation. This means that the key, and all its contents and sub-keys, will be removed regardless of whether other software information that is unrelated to this MSI exists. This can have a severe impact on other programs; only select this option if you are sure that the only software affected is the base MSI.
Both Create and Delete	Both of the above scenarios will occur.

Your new key is created with an empty default string value. To modify the value name and data, see [Creating a Registry Value](#).

6. To remove the key, right-click on it and select Delete.

Creating a Registry Value

Adding a New Value Name



Task: *To add a new value name:*

1. Select Registry from the checklist. The Registry View opens.
2. Select the key to which you want to add a value from the Destination Computer Registry view. Existing values for the key are displayed in the Destination Computer Registry Data view.
3. Right-click in the list of values and select New String Value, New Binary Value, or New DWORD Value, depending on the type of data you want to register.

A new empty value name is created with the name “New Value #n” (where *n* is a successive number).
4. Enter a meaningful name now to rename the value. To rename the value later, right-click on the value name and select Rename.



Note • When creating binary values, Tuner automatically converts whatever input you provide into a binary value.

Modifying the Value Data

Each new key has an empty default string value.



Task: *To modify this or any value data:*

1. Right-click on a value name and select Modify. The Edit Data dialog box opens.
2. In the Value data text box, enter a new value or edit the existing value.
3. Click OK.

Importing REG Files

Tuner allows you to import any existing REG files that you may have created previously. To import a REG file you need to launch the Registry Import Wizard.



Task: *To import a REG file:*

1. Select Registry from the checklist. The Registry View opens.
2. Right-click on a registry hive in the Destination Computer Registry View and select Import REG File. The Welcome panel of the Import REG File Wizard appears.
3. On the Welcome panel, click Next. The Import Registry File panel appears.

4. In the Registry File text box, either type the location of the registry file or browse to it, and click Next. The Import Conflict Options panel appears.
5. Select how you would like to handle duplicate registry data during the import. You have two options:

Option	Description
Overwrite the registry data	If any conflicts exist, the old registry keys will be overwritten by the new keys.
Do not overwrite the registry data	If duplicate keys are encountered, keep the existing keys.

6. After you have selected the method, click Import to continue. The Finishing Registry Import panel appears.
7. After the registry file has been scanned, click Finish to insert all entries from the REG file into the Destination Computer Registry view. You can then modify the entries.

Removing Registry Information



Task: *To remove registry information:*

1. Select Registry from the checklist. The Registry View opens.
2. Navigate to the registry entry that you want to remove in the Destination Computer Registry View.
3. If you want to remove a value from a specific key, right-click on the value in the Destination Computer Registry Data pane and select Delete.
4. If you want to remove an entire key, right-click on the key in the Destination Computer Registry View pane and select Delete or press the Delete key.

Shortcuts



The Shortcuts view offers an integrated, visual method for adding shortcuts and program folders to the installation. Existing shortcuts can also be modified or removed.

Shortcuts can be placed in:

- folders already defined by the installation,
- standard folders that are predefined by the Windows Installer such as the Fonts folder, or
- new folders which you can create.

Each shortcut has several properties that specify the target program, hot key combination, icon, and other information necessary to launch the application. When you create a new shortcut, it will always be installed.



Note • Shortcuts created in the transform are denoted by  and shortcuts from the base Windows Installer package are denoted by .

Topics in this section include the following:

- [Creating Shortcuts](#)
- [Changing a Shortcut's Icon](#)
- [Change a Shortcut's Location](#)
- [Changing a Shortcut's Target](#)
- [Creating a Hot Key](#)
- [Removing Shortcuts](#)
- [Determining the Path of Changed Shortcuts](#)

Creating Shortcuts



Task: *To create a shortcut:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. In the Shortcuts folder tree, navigate to the folder in which you want to put the shortcut.
3. Right-click on the folder and select New Shortcut.
4. Provide a name for the shortcut.
5. Enter properties for the shortcut in the Properties Grid.

Changing a Shortcut's Icon



Task: *To change the icon used for a shortcut:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. In the Shortcuts folder tree, navigate to the folder containing the shortcut you want to edit.
3. Select the Icon property from the Properties Grid.
4. Click the Change Icon button in the pane below the grid. The Change Icon dialog box opens.

5. Select one of the displayed icons or browse to the file that contains the icon you want to use for the shortcut.
6. After you have selected the appropriate icon, click OK. The new icon is now displayed to the left of the Change Icon button.

Change a Shortcut's Location



Task: *To change a shortcut's location:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. In the Shortcuts folder tree, navigate to the folder containing the shortcut you want to move.
3. Select the shortcut and drag it to another folder in the Shortcuts tree.

Changing a Shortcut's Target



Task: *To change a shortcut's target:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. In the Shortcuts folder tree, navigate to the folder containing the appropriate shortcut.
3. Select the Target property from the Properties Grid.
4. Select the appropriate Target from the list. The Target Type that is selected affects what you should enter in the Target property field:
 - **File from MSI Package & File from File System:** Provide the full path to the application or batch file.
 - **Destination Folder:** Select a folder name from the drop-down list. The list includes available folders on the target system, from the MSI package, and from the transform.
 - **Advertised Shortcut:** Enter the feature name. You can determine the name of the feature by going to the Direct Editor and selecting the Feature table. The list of features that you can target is listed in the Features column of the table.

Creating a Hot Key

A Hot Key is a combination of keys used to launch a shortcut instead of using the mouse.



Task: *To create a hot key:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. Select the shortcut to which you want to add the Hot Key.

3. Click on the Hot Key field in the properties grid. The Hot Key dialog box opens.
4. Press the keys on the keyboard that you want to use for the shortcut. The shortcut appears in the dialog.
5. If the shortcut is correct, click OK. The dialog box closes and the shortcut's converted ASCII value appears as the value for the Hot Key.



Note • These four fields in the Shortcuts Property Box are required for creating a Hot Key: Icon, Target, Run, and Hot Key. When you are creating the Hot Key Combination, DO NOT use a keyboard combination already adopted by Microsoft (such as Ctrl+V, which is used for Paste). Otherwise, the shortcut will not work.

Removing Shortcuts



Task: *To remove a shortcut:*

1. Select Shortcuts from the checklist. The Shortcuts View opens.
2. Use the Shortcuts Tree to navigate to the shortcut you want to delete.
3. Right-click on the shortcut and select Delete to remove it.

Determining the Path of Changed Shortcuts



Task: *To determine the actual path of a changed shortcut:*

1. Go to the Direct Editor and select the Directory table.
2. Find the shortcut Target directory, such as INSTALLDIR.
3. In that Directory row, find the value in the Directory_Parent column. In this example, the value is DIR26.
4. Look for a row in the Directory column that does not have a Directory_Parent entry. The directory in the row that has no value in the Directory_Parent column is the root directory. In this example, the root directory is TARGETDIR.

INI Files

Initialization (INI) files serve as a repository in which you can store and retrieve information between uses of your application. Typically INI Files contain key name-value pairs representing run-time options for applications. Some .ini files, such as Boot.ini and Wininit.ini, are used by the operating system.

INI files are divided into sections, each section containing keywords. Sections are divided by the square brackets surrounding them—[SectionName], for example. INI file keywords are the lowest level of organization in an .ini file. These keywords store data that must persist between uses of an application.

The INI Files View provides a graphical way for users to add, modify, or delete the contents of the IniFile Table. It displays the contents of the IniFile table from the source Windows Installer package and the transform.

Topics in this section include the following:

- [Adding INI Files](#)
- [Importing Existing INI Files](#)
- [Adding Sections to INI Files](#)
- [Adding New Keys to INI File Sections](#)
- [Modifying INI File Keys, Values, and Actions](#)
- [Removing INI Files](#)
- [Removing Sections from INI Files](#)
- [Removing INI File Section Keys](#)

Adding INI Files



Task: *To add an INI file to your transform file:*

1. Select INI Files from the checklist. The INI Files View opens.
2. Right-click on the appropriate destination folder in the INI File tree and select New IniFile.
3. To rename the new INI file, select IniFile#.ini and select Rename from the context menu.
4. To rename the new INI section, select NewSection#.ini and select Rename from the context menu.
5. With the new INI section selected, enter a Key name, Value, and Action for the default INI key value. See [Modifying INI File Keys, Values, and Actions](#).
6. Add additional Sections and Keys, as described in [Adding Sections to INI Files](#) and [Adding New Keys to INI File Sections](#).

Importing Existing INI Files



Task: *To import an existing INI file:*

1. Select INI Files from the checklist. The INI Files View opens.
2. Right-click on the appropriate destination folder in the INI File tree and select Import INI File from the context menu. The Welcome Panel of the Import INI File Wizard opens.
3. Click Next. The Import INI File Panel opens.
4. Enter or browse to the INI file you want to import. Click Next. The Import Conflict Options Panel opens.

5. Select how you want to handle duplicate keys and values. Click Import.
6. Once the INI file has been imported, click Finish.

The imported INI file appears under the selected destination folder. You can then make further adjustments to it as needed.

Adding Sections to INI Files



Task: *To add a section to an INI file:*

1. Select the INI Files View from the checklist.
2. Right-click on the appropriate INI File in the INI File tree and select New Section. A new INI section, named NewSection1, is created under the selected INI file, complete with a default Key name, Value, and Action.
3. To rename the new section, select NewSection1 and select Rename from the context menu.
4. With the new INI section selected, enter a Key name, Value, and Action for the default INI key value. See [Modifying INI File Keys, Values, and Actions](#).
5. Add additional Keys to this new INI file section as necessary. See [Adding New Keys to INI File Sections](#).

Adding New Keys to INI File Sections



Task: *To add a new INI file key:*

1. Select the INI Files View from the checklist.
2. Expand the listing of the INI file that you would like to edit so that all of its sections are displayed.
3. Select the INI file section that you would like to edit. That section's defined keys are listed on the right.
4. Right-click in the key listing and select Add from the menu. A new key is added to the key listing with a default Key name, Value and Action.
5. Edit the Key name, Value, and Action for this new Key. See [Modifying INI File Keys, Values, and Actions](#).

Modifying INI File Keys, Values, and Actions

INI files contain key name-value pairs representing run-time options for applications. To define and modify the key names, values, and /or actions, perform the following steps.



Task: *To modify keys, values, and/or actions in INI files:*

1. Select the INI Files View from the checklist.
2. In the INI File tree, select the INI file that you want to edit, and expand the listing so that you can select the appropriate section. That section's keys, values, and actions are listed:

Destination Computer

ProgramFilesFolder

InstallShield

AdminStudio

5.0

AdminStudio.ini

Common

Developer.ini

developer

Repackager

developer

Key	Value	Action
Author	Y	Add Line
Title	InstallShield Developer - AdminStudio Edition	Add Line
Ver	Includes 8.01	Add Line
ConflictServices	[INSTALLDIR]Common\ISUIConflictServices.c	Add Line
FileOpen	[INSTALLDIR]Common\ASFileOpen.dll	Add Line
ISToday	[INSTALLDIR]Developer\Program\0409\ISADI	Add Line
HelpFile	[CommonFilesFolder]InstallShield\AdminStudi	Add Line
Extensions	Repackager Output File (*.inc) *.inc, ZENwv	Add Line
EvalCheck	[INSTALLDIR]Common\ISCommonHelper.dll	Add Line
InstallLocation	[INSTALLDIR]Developer	Add Line
NavFile	[INSTALLDIR]Developer\Program\0409\ispro	Add Line

3. Modify the Key name, Value, and Action for each key, as necessary.

Option	Description
Key	The name of the key. This should be entered in the exact way you want it to appear in the target INI file.
Value	The key's value. Windows Installer properties can be used in your keyword's value. To do this, surround the property with square brackets—[INSTALLDIR], for example. For a comprehensive list of Windows Installer properties, refer to the Property Reference topic in the Windows Installer help .
Action	The action the key performs. Select this from the list in the property sheet. The available options are: <ul style="list-style-type: none">• Add Tag: Creates a new entry or appends a new comma-separated value to an existing entry.• Create Line: Creates a .ini entry only if the entry does not already exist.• Add Line: Creates or updates a .ini entry.

Removing INI Files



Task: *To remove an INI file:*

1. Select the INI File view from the checklist.
2. From the INI File tree, right-click on the INI file you want to delete and select Remove.

Removing Sections from INI Files



Task: *To remove a section from INI file:*

1. Select the INI File view from the checklist.
2. From the INI File tree, right-click on the section you want to delete and select Remove.

Removing INI File Section Keys



Task: *To remove an INI file section key:*

1. Select the INI Files view from the checklist.
2. Select the INI File section that contains the key you want to delete from the INI Files tree.
3. Right-click the key you want to remove and select Delete.

ODBC Resources

Open Database Connectivity (ODBC) Resources are ones that involve interaction with databases. Tuner allows you to view existing ODBC Data Sources, ODBC Drivers, and ODBC Translators.

Topics in this section include the following:

- [Adding New Data Sources](#)
- [Adding New ODBC Data Source Attributes](#)
- [Adding New ODBC Driver Attributes](#)
- [Editing ODBC Data Source Attributes](#)
- [Editing ODBC Driver Attributes](#)
- [Removing Existing ODBC Data Sources](#)
- [Removing ODBC Driver Attributes](#)
- [Removing ODBC Data Source Attributes](#)

Adding New Data Sources



Task: *To add a new ODBC Data Source:*

1. Select ODBC Resources from the checklist. The ODBC Resources View opens.
2. Right-click either ODBC Data Sources group or one of its children groups from the ODBC Resources tree and select New Data Source from the context menu. The ODBC Data Source dialog box opens.
3. Select the required data source and click OK.



Caution • If you are adding an ODBC Data Source that does not exist on your computer, type the name of the Data Source into the ODBC Data Source dialog. Keep in mind that adding a data source to a Windows Installer package that does not contain the corresponding driver may render the package useless.

Adding New ODBC Data Source Attributes



Task: *To add a new ODBC data source attribute:*

1. Select ODBC Resources from the checklist. The ODBC Resources View opens.
2. Select the ODBC Data Source to which you want to add a new attribute from the ODBC Resources tree. The property grid for the selected ODBC Data source opens.
3. Right-click in the property grid and select Add. A new attribute is listed, with the default values of ATTRIBUTE and NULL_VALUE.
4. Enter information for the new attribute.

Adding New ODBC Driver Attributes



Task: *To add a new ODBC driver attribute:*

1. Select ODBC Resources from the checklist. The ODBC Resources View opens.
2. Select the ODBC driver to which you want to add a new attribute from the ODBC Resources tree. The property grid for the selected ODBC driver appears.
3. Right-click in the property grid and select Add from the context menu. A new attribute is listed, with the default values of ATTRIBUTE and NULL_VALUE.
4. Enter information for the new attribute.

Editing ODBC Data Source Attributes



Task: *To edit an ODBC data source attribute:*

1. On the ODBC Resources View, select the ODBC data source that contains the attribute you want to modify from the ODBC Resources tree. The property grid for that data source appears.
2. In the properties grid, edit the appropriate attribute.

Editing ODBC Driver Attributes



Task: *To edit an ODBC driver attribute:*

1. On the ODBC Resources View, select the ODBC driver that contains the attribute you want to modify from the ODBC Resources tree. The property grid for that ODBC driver appears.
2. In the properties sheet, edit the appropriate attribute.

Removing Existing ODBC Data Sources



Task: *To remove an existing ODBC Data Source:*

1. Select the ODBC Resources view from the checklist.
2. Right-click on the ODBC data source you want to remove from the ODBC Resources tree and select Delete.

Removing ODBC Driver Attributes



Task: *To remove an ODBC driver attribute:*

1. Select the ODBC Resources view from the checklist.
2. Select the ODBC driver that contains the attribute you want to delete from the ODBC Resources tree.
3. In the properties sheet, right-click on the attribute you want to remove and select Delete.

Removing ODBC Data Source Attributes



Task: *To remove an ODBC data source attribute:*

1. Select the ODBC Resources view from the checklist.
2. Select the ODBC data source that contains the attribute you want to delete from the ODBC Resources tree.
3. In the properties sheet, right-click on the attribute you want to remove and select Delete.

NT Services

The NT Services view provides a way to change parameters for NT Services included in the base Windows Installer package. Topics in this section include the following:

- [Setting NT Service Arguments](#)
- [Setting NT Service Dependencies](#)
- [Setting the NT Service Description](#)
- [Setting the NT Service Display Name](#)
- [Setting the NT Service Error Control Level](#)
- [Setting the NT Service Load Order Group](#)
- [Setting the NT Service Overall Install Result](#)
- [Setting the NT Service Start Type](#)
- [Setting NT Service Start Name and Password](#)
- [Setting the NT Service Type](#)

Setting NT Service Arguments



Task: *To set NT service arguments:*

1. Select the NT Services from the checklist. The NT Services View opens.
2. Double-click the current value in the properties grid and modify it as needed.

Setting NT Service Dependencies



Task: *To set NT service dependencies:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Dependencies value in the properties grid and add the names of services or load ordering groups that must be started prior to this service.



Note • *If the dependency is on a load ordering group, the service can start if at least one member of the load ordering group is running after an attempt is made to start all load ordering group members.*

Setting the NT Service Description



Task: *To set the NT service description:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Description value in the properties grid and modify it as needed.

Setting the NT Service Display Name



Task: *To set the NT service display name:*

1. Select the NT Services view from the checklist.
2. Double-click the current Display Name value and modify it. The display name can be up to 256 characters in length.

Setting the NT Service Error Control Level



Task: *To set the NT service error control level:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Error Control value in the properties grid to access the pop-up menu. The possible values are as follows:

Value	Description
Ignore Error	Logs the error and continues with service startup.
Normal Error	Logs the error, displays an error message, and continues with service startup.
Critical Error	Logs the error (if possible) and restarts the system with the last configuration known to be good. If the last-known-good configuration is the one that caused the error, fail the startup.

Setting the NT Service Load Order Group



Task: *To set the NT service load order group:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Load Order Group value in the properties grid and modify it as needed. If this service does not belong to a group, leave this value blank.

Setting the NT Service Overall Install Result



Task: *To set the NT service overall install result:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Overall Install value in the properties grid to access the drop-down menu.
3. Select either Continue overall install if service fails to install or Fail overall install if service fails to install as this property's value.

Setting the NT Service Start Type



Task: *To set the NT service start type:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Start Type value in the properties grid to access the drop-down menu.
3. Select the desired Start Type from the following possible values:

Option	Description
Automatic	The service starts during system startup.
Manual	The service is only started when the service control manager calls the StartService function.
Disabled	The service is not started.
Start at Boot Time	The driver is started by the operating system loader. (Device driver only)
Started by the System	The driver is started by calling the IoInitSystem function. (Device driver only)

Setting NT Service Start Name and Password



Task: *To set the NT service start name and password:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Start Name value in the properties grid and provide the name under which this service will run.
3. Click the current Password value in the properties grid and provide the password associated with the Start Name.

Setting the NT Service Type



Task: *To set the NT service type:*

1. Select NT Services from the checklist. The NT Services View opens.
2. Double-click the current Service Type value in the properties grid to access the drop-down menu.
3. Select the desired Start Type from the following possible values:

- Service that Runs in its Own Process
 - Service that Shares a Process with Others
4. Optional: click the current Interact with Desktop value in the properties grid to access the drop-down menu and specify whether the service needs to interact with the desktop.

Working with Dialogs

When customizing the Windows Installer package, you may want to disable particular panels that appear during the installation, administrative, patch, or maintenance sequences. You can do so from the Dialogs view. This view contains a list of each of the four installation modes (installation, administrative, maintenance, and patch), with the associated dialogs that appear as part of the UI sequence during the selected mode. You can enable or disable these dialogs by either the check box to the left of the dialog name, or by using the Show and Hide buttons.

Topics in this section include the following:

- [Hiding Dialogs During UI Sequences](#)
- [Restoring Dialog Sequences](#)
- [Suppressing the License Agreement Dialog Box](#)
- [Disabling Custom Setups](#)
- [Editing Dialog Properties](#)
- [Dialogs View vs. Command-Line Options](#)
- [Dialog Suppression Issues](#)

Hiding Dialogs During UI Sequences



Task: *To hide a dialog box during a UI sequence:*

1. Select Dialogs from the checklist. The Dialogs View opens.
2. Using the Installation Mode list, select the UI sequence containing the dialog you want to hide.
3. From the Dialogs list, clear the check mark next to the dialog you want to hide. Alternatively, select the dialog from the list and either click Hide or press the Space Bar.
4. If necessary set the properties for the dialog you are removing to preserve the UI sequence integrity. The Dialog Properties dialog box automatically appears if it is necessary to edit properties.

Restoring Dialog Sequences



Task: *To restore dialogs in a UI sequence:*

1. Select Dialogs from the checklist. The Dialogs View opens.
2. Using the Installation Mode list, select the UI sequence containing the dialog you want to restore.
3. From the Dialogs list, check the box next to the dialog you want to restore. Alternatively, select the dialog from the list and either click Show or press the Space Bar.

Suppressing the License Agreement Dialog Box

Using the Dialogs view, it is possible to suppress the license acceptance dialog. This involves both turning off its display, and providing the value that the setup will interpret as acceptance of the agreement.



Note • This procedure assumes the original Windows Installer setup was created using InstallShield Editor. If another setup authoring application was used, the names of dialogs and properties may not be the same. The same general procedure still applies.



Task: *To use a transform to suppress display of the license acceptance dialog:*

1. Select Dialogs from the checklist. The Dialogs View opens.
2. Using the Installation Mode list, select the UI sequence containing the License Agreement dialog. Typically, this is only in the Installation Sequence.
3. Select the **LicenseAgreement** dialog from the Dialogs list and click Hide.
4. When the Dialog Properties dialog box appears, change the AgreeToLicense value to Yes.
5. Click OK to dismiss the dialog.

When an end user runs the installation using this transform, the License Agreement dialog will not appear.

Disabling Custom Setups

You can use the Dialogs view to prevent users from performing custom setups. However, this requires not only the elimination of the Custom Setup panel during installation, but also the Setup Types panel. Additionally, you must ensure you have configured the features you want installed, as your end user will have no way to override them.



Note • This procedure assumes the original Windows Installer setup was created using InstallShield Editor. If another setup authoring application was used, the names of dialogs and properties may not be the same. The same general procedure still applies.



Task: *To use a transform to disable a custom setup:*

1. Select Dialogs from the checklist. The Dialogs View opens.
2. Using the Installation Mode list, select the UI sequence containing the SetupType dialog. Typically, this is only in the Installation Sequence.
3. Select the SetupType dialog from the Dialogs list and click Hide.
4. When the Dialog Properties dialog box appears, change the ADDLOCAL value to ALL.
5. Click OK to dismiss the dialog.
6. Select the CustomSetup dialog from the Dialogs list and click Hide.
7. When the Dialog Properties dialog box opens, change the _BrowseProperty value to INSTALLDIR.
8. Click OK to dismiss the dialog.

When an end user runs the installation using this transform, the user will not have the option to perform a custom setup.

Editing Dialog Properties



Task: *To edit properties for a UI sequence dialog:*

1. Select Dialogs from the checklist. The Dialogs View opens.
2. Using the Installation Mode list, select the UI sequence containing the dialog box containing properties you want to edit.
3. From the Dialogs list, select the appropriate dialog.
4. Click Properties. The Dialog Properties dialog box opens.



Note • The Properties button is only enabled when you have selected a dialog box containing editable properties.

5. From the Dialog Properties dialog box, double-click the value cell for the property you want to edit.
6. Change the property value as necessary, and click OK.

Dialogs View vs. Command-Line Options

Generally, you should use the Dialogs View when you are still planning on displaying some panels during UI sequences. Typically, you may want to remove the License Agreement panel or the ability for a user to perform a custom setup, and these can both be accomplished easily from the Dialogs View.

However, consider using the Windows Installer command-line options (particularly `/qn`) when you want to eliminate the user interface entirely.

Dialog Suppression Issues

When suppressing dialog box display, it is important to consider some implications of your actions. Particularly, when removing a dialog from a user interface sequence, there may be properties normally set by that dialog. For example, the LicenseAgreement dialog has a radio button which can set a property to Yes or No, depending on whether you agree to the terms in the license. The value of this property also determines whether the installation should continue. Therefore, if you remove the LicenseAgreement dialog from a sequence, you need to use the Dialog Properties dialog box to set the value of this property so the installation can continue.

Beyond setting necessary properties, you also should consider how features are displayed. For example, you may want to disable custom setups via the transform file. However, you must ensure each feature you want installed is configured to be installed; your end users will have no way to override the choices by performing a custom setup.

Configuring Additional Server Locations

If you install from a network server, and if you install features to run from the server or that will be advertised for installation on their first use, the applications may need access to the server sometime after the initial installation. The applications may also require access to the server if a file is deleted or becomes corrupt, as the application can copy the problematic file(s) automatically from the server.


Topics in this section include the following:

- [Adding Additional Server Locations](#)
- [Modifying Server Locations](#)
- [Removing Server Locations](#)
- [Reordering Server Locations](#)

Adding Additional Server Locations



Task: *To add an additional server location:*

1. Select Server Locations from the checklist. The Server Locations View opens.
2. Click the New button () in the Addition Server Location Paths window.
3. Enter or click the Browse (...) button and browse to the server location.



Note • The validity of the server location is determined when the installation needs to access the server remotely. In other words, if a server is not available, or if you added an invalid server, the entry will be ignored if the resource is needed.

Modifying Server Locations




Task: *To modify an additional server location entry:*

1. Select Server Locations from the checklist. The Server Locations View opens.
2. Select the server entry from the Addition Server Location Paths window and either edit the entry or use the Browse (...) button to browse to desired location.

Removing Server Locations




Task: *To remove an additional server location:*

1. Select Server Locations from the checklist. The Server Locations View opens.
2. Select the server entry from the Addition Server Location Paths window.
3. Click the Remove button ()

Reordering Server Locations



Task: *To change the order in which additional server locations are accessed:*

1. Select Server Locations from the checklist. The Server Locations View opens.
2. Select a server entry from the Addition Server Location Paths window.
3. Depending on whether you want to promote the server location or demote it, click the up and down arrow buttons at the top right of the view (). You can also use the Alt+Up Arrow and Alt+Down Arrow keyboard shortcuts.
4. Repeat with other server location entries as necessary.

Changing Add/Remove Program Settings

The Windows 2000 and Windows XP Add/Remove Programs in Control Panel differs from the previous Windows operating systems in many ways. Depending on how the Windows Installer setup is configured, the user has the option of removing, repairing, or changing the installation with the click of a button.

Windows 2000 and XP users are also able to access additional information in Add/Remove Programs in Control Panel not available on previous platforms. With this information, it is easier for your end users to find technical support links and phone numbers, product update information, and information about your company.

Topics in this section include the following:

- [Changing Add/Remove Programs Properties](#)
- [Disabling the Modify, Remove, or Repair Buttons](#)

Changing Add/Remove Programs Properties



Task: *To change properties in the Windows 2000 or XP Add/Remove Programs:*

1. Select Add/Remove Programs from the checklist. The Add/Remove Programs View opens.
2. Double-click the value for the property you want to change.
3. Either enter the information into the properties grid, or use the drop-down menu to select a value.

Disabling the Modify, Remove, or Repair Buttons



Task: *To disable the Modify, Remove, or Repair buttons in Windows 2000 or XP's Add/Remove Programs in Control Panel:*

1. Select Add/Remove Programs from the checklist. The Add/Remove Programs View opens.
2. Double-click the appropriate Disable Modify/Remove/Repair Button property from the properties grid.
3. Use the drop-down menu to change the value to No.

Customizing Setup Properties

Even though Tuner provides you views to customize many areas of the Windows Installer package, it may be necessary to edit property values that are not available elsewhere. The Setup Properties view exposes the entries in the properties table (the underlying structure of Windows Installer packages). You can also add your own custom properties here.

Topics in this section include the following:

- [Adding Custom Setup Properties](#)
- [Adding and Editing Comments](#)
- [Removing Custom Setup Properties](#)
- [Modifying Setup Properties](#)

Adding Custom Setup Properties



Task: *To add a new setup property:*

1. Select Setup Properties from the checklist. The Setup Properties View opens.
2. Right-click in the properties grid and select Add. A New property is added to the bottom of the list with the a Property Name of NEW_PROPERTY and a Value of NULL_VALUE.
3. Provide a new name for your property. If you want to change it later, click on the property name to edit it.
4. Enter the property's value.

Adding and Editing Comments

Tuner supports the ability to add comments to each property available in the Setup Properties view. This provides a way to clarify what specific properties do, and to enter any important information that you may need later. The original software vendor may have used InstallShield Editor to include comments in the original Windows Installer package.



Task: *To add or edit a comment for a property:*

1. Select the Setup Properties view from the checklist.
2. Double-click in the comment column for the property to which you want to add or edit the comment.
3. Add or edit the comment as appropriate.

Removing Custom Setup Properties



Task: *To remove a custom setup property:*

1. Select Setup Properties from the checklist. The Setup Properties View opens.
2. Right-click on the property you want to remove and select Delete.
3. Confirm the deletion.

Modifying Setup Properties



Task: *To modify the property value of a setup:*

1. Select Setup Properties from the checklist. The Setup Properties View opens.
2. Select the property that you want to modify.
3. Double-click the property's value and edit it in the grid.

Make sure you are entering valid values when you modify properties, otherwise validation or installation errors may result.

Preparing Packages for Distribution

The final step in creating a customization involves two parts. First, you should postvalidate your transform and base Windows Installer package. This ensures that you have not introduced any errors into the installation, and may help you verify that you have corrected errors that existed in the base package. Secondly, you need to actually package the transform and base package for distribution. These tasks are accomplished using the Postvalidation view and Package view, respectively.

Topics in this section include the following:

- [Copying the Installation to a Network Location](#)
- [Copying the Installation to an FTP Server](#)
- [Creating a Package Definition File \(PDF\)](#)
- [Creating an SMS File](#)
- [Instructing SMS to Create a Management Information Format File at Deployment Time](#)
- [Deploying Windows Installer Setup Packages with Systems Management Server 2.0](#)
- [Creating a Setup.exe File for the Package and Transform](#)
- [Additional Setup.ini Parameters](#)

Copying the Installation to a Network Location



Task: *To copy your installation to a Network location during packaging:*

1. From the checklist, select Package, and then select Location from the second column. The Location View of the Package View opens.
2. Select the Network Location option button.
3. Specify the network location, or click Browse to locate it.

When you create your package, the appropriate files will be copied to the network location you specified.



Note • *If the transforms are copied to the same location as the original MSI, only the transform, setup.exe, setup.ini, and Windows Installer engines are copied.*

Copying the Installation to an FTP Server



Task: *To copy your installation to an FTP server during packaging:*

1. From the checklist, select Package, and then select Location from the second column. The Location View of the Package View opens.
2. Select the FTP Server option button.
3. Specify the FTP server name (FTP Location), the UserName, and Password for the FTP server.

When you create your package, the appropriate files will be copied to the FTP server you specified.

Creating a Package Definition File (PDF)



Task: *To create a Package Definition File (PDF):*

1. From the checklist, select Package, and then select SMS from the second column. The SMS View of the Package View opens.
2. Select the Create Package Definition File check box.

When you create your package, the resulting file has a .PDF extension. Here is a sample ORCA.PDF file:

```
[PDF]
Version=1.0

[Package Definition]
Product=Orca
Version=1.20.1827.1
Comment=Microsoft
SetupVariations=Typical, Automated

[Typical Setup]
CommandName = Typical Installation
CommandLine = msixexec /i Orca.msi
UserInputRequired = TRUE
SynchronousSystemExitRequired = TRUE
SupportedPlatforms = Win 9x, Win NT (i386)

[Automated Setup]
CommandName = Automated Installation
CommandLine = msixexec /i /q Orca.msi
UserInputRequired = FALSE
SynchronousSystemExitRequired = TRUE
SupportedPlatforms = Win 9x, Win NT (i386)

[Setup Package for Inventory]
InventoryThisPackage=FALSE
```

Creating an SMS File



Task: *To create a SMS file for SMS 2.0 or later:*

1. From the checklist, select Package, and then select SMS from the second column. The SMS View of the Package View opens.
2. Select the Create SMS file check box.

When you create your package, the resulting file has an .SMS extension. Here is a sample ORCA.SMS file:

```
[PDF]
Version=2.0
```

```
[Package Definition]
MIFFilename=Sample.MIF
Name=Orca
Publisher=Microsoft
Version=1.20.1827.1
Language=English
Programs=Typical, Automated, Test
```

```
[Typical]
Name = Typical
CommandLine = msixexec /i Orca.msi
UserInputRequired = TRUE
UninstallKey={8FC71000-88A0-4B41-82B8-8905D4AA904C}
AfterRunning=ProgramRestart
SupportedClients = Win 9x, Win NT (i386)
```

```
[Automated]
Name = Automated
CommandLine = msixexec /i /q Orca.msi
UserInputRequired = FALSE
UninstallKey={8FC71000-88A0-4B41-82B8-8905D4AA904C}
AfterRunning=ProgramRestart
SupportedClients = Win 9x, Win NT (i386)
```

```
[Test]
Name = Test
CommandLine = msixexec /i Orca.msi EXECUTEMODE=None
UserInputRequired = FALSE
AfterRunning=ProgramRestart
SupportedClients = Win 9x, Win NT (i386)
```

Instructing SMS to Create a Management Information Format File at Deployment Time



Task: *To instruct SMS to create a Management Information Format (MIF) file at deployment time:*

1. From the checklist, select Package, and then select SMS from the second column. The SMS View of the Package View opens.
2. Select the Create SMS file check box.
3. Provide the name of the application you are installing in the Install MIF Filename field.
4. Provide the name of the application to be uninstalled in the Uninstall MIF Filename field.
5. Enter the serial number for the product in the Serial Number field.

The resulting file has a .MIF extension.

Deploying Windows Installer Setup Packages with Systems Management Server 2.0



Task: *To deploy Windows Installer setup packages with SMS 2.0:*

Perform the steps detailed in the Microsoft White Paper: [Deploying Windows Installer Setup Packages with Systems Management Server 2.0](#).

Creating a Setup.exe File for the Package and Transform



Task: *To create a Setup.exe file to begin the installation of your base package and transform:*

1. From the checklist, select Package, and then select Setup from the second column. The Setup View of the Package View opens.
2. Select the Create Installation Launcher (Setup.exe) check box.
3. Specify whether you want to include the Windows 95/98 or Windows NT MSI engines.
4. Provide any command-line arguments for your installation.
5. Save your transform. The Setup.exe file is stored in the directory specified in the Location panel of the Packaging Wizard, or in the Location view within the Package view.

Additional Setup.ini Parameters

You can modify the Setup.ini file generated by Tuner for added functionality. The following two parameters can be included in the [Startup] section of the Setup.ini:

Table 12-3 • Setup.ini File [Startup] Section Parameters

Parameter	Description
SuppressWin2k	If you add this line and set its value to “Y” or “y” (SuppressWin2k=Y or SuppressWin2k=y), Setup.exe will not display a message on the target system stating that an older version of Windows Installer is installed.
SuppressReboot	If you add this line and set its value to “Y” or “y” (SuppressReboot=Y or SuppressReboot=y), Setup.exe will delay the reboot typically required by the installation of a newer version of Windows Installer on the target machine. If the application's setup requires a reboot, this will occur normally.

Directly Editing Packages

Windows Installer packages are relational databases consisting of dozens of interrelated tables. These tables reflect the application's features, components, relationship between features and components, registry information, and user interface.

The Direct Editor allows you to edit values in the MSI tables of the base Windows Installer package and store them in your transform. As you change values elsewhere in your transform, those changes are reflected in the Direct Editor, and vice versa. The complete list of MSI tables contained in the installation package is displayed in the left pane. When you select a table, the contents are displayed in the right pane.

Resizing Table Columns in the Direct Editor

When you initially open the Direct Editor, the selected table's columns are listed in a compact format so that the maximum number of columns are displayed.

To automatically resize a column so that its width matches that of its longest entry, double-click on the column heading. This new column width setting is automatically saved and will be implemented the next time you view this table column in the Direct Editor.

Sorting Table Columns in the Direct Editor

To sort a table column, click the column heading once. The order will toggle between ascending and descending.

Adding a New Record Using the Direct Editor



Task: *To add a new record in the Direct Editor:*

1. Select Direct Editor from the checklist. The Direct Editor opens.
2. Select the table you want to add a row to from the table tree.
3. Click on the last row in the table's grid, or click anywhere in the table and press the Insert key.

A new record is added to the grid using a unique key for the record (use ColumnName<n> as the template). You can then modify the record.



Caution • To ensure files are deployed on the target system, add files in the Files and Folders View, rather than to the file table. Files added to the File table are not physically added to the transform.

Finding and Replacing Using the Direct Editor

The Direct Editor supports find and replace throughout all tables. The following commands are available:

Table 12-4 • Direct Editor Commands

Command	Description
Find (Ctrl+F)	Displays a standard Find dialog box and allows you to search for a string in all tables.
Find Next (F3)	Allows you to search for the next occurrence of a given string in all tables.
Replace (Ctrl+H)	Displays a standard Search and Replace dialog box and allows you to search for data in all the MSI tables and gives you the option to replace the data.

Launching the Direct Editor from the Validation Tab

When performing a Prevalidation or Postvalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (✖), a Warning (⚠), or an Informational Message (ℹ), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row's associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red. This feature makes it very easy for you to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform.

Resizing Table Columns in the Direct Editor

When you initially open the Direct Editor, the selected table’s columns are listed in a compact format so that the maximum number of columns are displayed.

To automatically resize a column so that its width matches that of its longest entry, double-click on the column heading. This new column width setting is automatically saved and will be implemented the next time you view this table column in the Direct Editor.

Sorting Table Columns in the Direct Editor

To sort a table column, click the column heading once. The order will toggle between ascending and descending.

Tuner Reference

Reference information for Tuner is divided into the following sections:

Table 12-5 • Tuner Reference Sections

Section	Description
User Interface Reference	Contains general information about the Tuner user interface, including menus, toolbars, keyboard shortcuts, windows, and dialog boxes.
Tuner Views	Contains detailed reference information about each Tuner view.
Import INI File Wizard	Provides reference information for the Import INI File Wizard. This is the same information accessible by clicking Help in the Wizard.
Import REG File Wizard	Provides reference information for the Import REG File Wizard. This is the same information accessible by clicking Help in the Wizard.
Packaging Wizard	Contains panel-by-panel reference information for the Tuner Packaging Wizard. Also includes information about additional Setup.ini options.

User Interface Reference

This book describes the user interface components, such as menu items, the toolbar, views, and dialog boxes you will encounter throughout Tuner.

- [Menus and Toolbar](#)
- [View Bar](#)
- [Checklist](#)
- [Output Window](#)
- [Customize Dialog Box](#)

- [Properties Dialog Box](#)

These topics are the same detailed documentation that is displayed when you press the F1 key or click the Help button while working in a dialog.

Menus and Toolbar

The Tuner user interface has several menus, each of which contain different commands. The functionality of each command is described below. Additionally, the Toolbar provides quick access to some of the frequently used commands; the corresponding toolbar buttons are listed with the appropriate commands.

Table 12-6 • Tuner Menus and Toolbar









Menu	Command	Shortcut	Toolbar Button	Description
File	New	Ctrl+N		Takes you to the Create a New Transform File area of the Tuner Start Page view.
File	Open	Ctrl+O		Takes you to the Open a Recent Transform File area of the Tuner Start Page view.
File	Close			Closes the currently open transform file.
File	Save	Ctrl+S		Saves the current transform file.
File	Save As			Prompts you to name the transform file you are saving.
File	Properties			Displays properties for the current transform, including the name and location of the base Windows Installer package.
File	[1], [2], [3], or [4]			Allows you to select one of the four most recently accessed transforms.
File	Exit			Exits Tuner.
Edit	Undo	Ctrl+Z		Undoes the last action.
Edit	Cut	Ctrl+X		Removes the selected text to the clipboard.
Edit	Copy	Ctrl+C		Copies the selected text to the clipboard.
Edit	Paste	Ctrl+V		Pastes the contents of the clipboard to the current cursor location.
View	Output Window			Toggles the Output window.

Table 12-6 • Tuner Menus and Toolbar (cont.)

Menu	Command	Shortcut	Toolbar Button	Description
View	Check List			Toggles the checklist.
View	View Bar			Toggles the View Bar.
View	Header Bar			Toggles the Header Bar.
View	Toolbar			Toggles the Toolbar.
View	Status Bar			Toggles the Status Bar.
Go	Previous View			Takes you to the previous view in the checklist.
Go	Next View			Takes you to the next view in the checklist.
Go	Back			Moves you to the last view.
Go	Forward			Moves you to the next view.
Go	Start Page			Takes you to the Tuner Start Page.
Go	Help			Takes you to the Help view.
Go	Package Validation			Takes you to the Prevalidation view.
Go	Organization			Takes you to the Product Properties and Features views.
Go	System Configuration			Takes you to the Files, Registry, Shortcuts, INI Files, NT Services, and ODBC Resources views.
Go	Application Configuration			Takes you to the Server Locations, Setup Properties, Dialogs, and Add/Remove Programs views.
Go	Package Preparation			Takes you to the Postvalidation and Package views.
Go	Additional Tools			Takes you to the Direct Editor view.
Project	MSI Validation			Runs prevalidation on the MSI file.

Table 12-6 • Tuner Menus and Toolbar (cont.)

Menu	Command	Shortcut	Toolbar Button	Description
Project	Transform Validation			Runs postvalidation on the MST file.
Project	Transform Summary Information			Launches the Transform Summary dialog.
Project	Test	Ctrl+T		Allows you to test your custom installation without actually installing.
Project	Run	Ctrl+F5		Performs the actual installation of the Windows Installer package and your transform.
Project	Package			Packages the transform and Windows Installer package based on the current packaging settings.
Project	Packaging Wizard	Ctrl+F7		Launches the Packaging Wizard.
Project	Stop			Halts in-progress validation or packaging.
Tools	Customize			Allows you to customize toolbars and menus.
Tools	Options			Allows you to specify the default locations for MSI and MST files.
Help	Help Library			Launches the online help (which you are currently viewing).
Help	MSI Help			Brings up the MSI online help.
Help	ReadMe			Displays the AdminStudio ReadMe file.
Help	Support Central			Connects to the AdminStudio Support Web site.
Help	About Tuner			Displays information about Tuner, including the version and copyright notice.

View Bar

The View Bar, when visible, is located at the far left of the user interface. It provides quick shortcuts to important areas of Tuner, and can be toggled on and off using the View Bar command under the View menu, or from the corresponding toolbar button.

There are three different View Bars available:

InstallShield

This View Bar gives you quick access to the Tuner Start Page and the Help view.

Checklist Steps

This View Bar provides access to each of the checklist steps, which include Package Validation, Organization, System Configuration, Application Configuration, Package Preparation, and Additional Tools views.

Views

This View Bar gives you quick access to each view in Tuner.

Checklist

The checklist is a graphical tree that shows you all of the views available in Tuner, as well as their association with other views. When you select a view, it appears in the pane to the right of the checklist; you can then customize the part of the Windows Installer package pertaining to that view. The Customization Steps Checklist is a subset of the entire checklist.

The checklist can be toggled from the View menu and from the Toolbar.

Customization Steps Checklist

To assist you with your customization, Tuner provides you with a set of steps that cover all parts of the MSI file that can be customized in your transform project. You do not have to follow these steps in sequential order, or even complete all the steps. Below is a brief description of each step, and how they fit into the customization workflow.

Table 12-7 • Customization Steps

Step	Description
Prevalidation View	This step allows you to pre-check the base Windows Installer package to ensure it is valid according to MSI standards before you take the time to create the transform file. If it is invalid, there may be unexpected results with your installation. Tuner allows you to copy the results of the prevalidation to the clipboard, where you can paste them into a message to the application's vendor. This is one of the most important steps in the Tuner workflow. Without it, all of your work might be wasted.

Table 12-7 • Customization Steps (cont.)

Step	Description
Organization View	This allows you to specify both the default destination that the installation will suggest, as well as the suggested default company name. You also can modify some properties of individual features in the product.
System Configuration View	This is where Tuner shows you its true customization power. You can add files and registry entries to the installation, allowing you to add company-specific files, templates, etc. to the installation. For example, if you have a set of word processing templates your company uses, you can include them you can modify or remove existing shortcuts, or add your own as necessary.
Application Configuration View	This step allows you to change functionality for Add/Remove Programs in Control Panel for Windows 2000 and later. You can also see the full spectrum of MSI properties as they exist in the base Windows Installer package, and add, modify, or remove them as necessary. Additionally, you can configure source resiliency through the Server Locations view, and customize user interface sequences through the Dialogs view.
Package Preparation View	This step allows you to postvalidate your project, verifying the original Windows Installer package and transform file, meets MSI-validation standards. You also can select the distribution options, including ones for network, FTP, single-executable, and SMS distributions.

Output Window

When you prevalidate the base Windows Installer package, postvalidate the package and your transform, or package the transform and base package, the Output Window appears at the bottom of the interface. It consists of three tabs:

- **Output**—Lists the Errors (❌), Warnings (⚠️), and Informational Messages (ℹ️) that are generated during prevalidation, postvalidation, and packaging.
- **Validation**—Upon completion of the pre- or postvalidation, this tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (❌), a Warning (⚠️), or an Informational Message (ℹ️), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row's associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red. You can then use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform.

- **Packaging**—Displays packaging/distribution information, and displays a summary of the files copied. You can copy these results to the clipboard by right-clicking and selecting Copy.

The Output Window can be toggled from the View menu.

Customize Dialog Box

The Customize Dialog box allows you to customize which toolbars are available in the Tuner user interface, as well as the buttons that are available on the toolbars. The dialog box consists of two tab panels:

Toolbars

From the Toolbars panel, you can select viewing properties for all toolbars, such as whether tooltips are displayed, the style of the toolbar, and the size of the buttons. You can also create your own custom toolbar, onto which you can place buttons found in the Command tab panel.

Command

The Command panel allows you to customize toolbars and the menu bar. Simply drag the command or menu you want to add to the existing toolbar; it appears where you place it. To remove a command or menu, select it and drag it off the toolbar.

Properties Dialog Box

This dialog box displays properties of the transform you are currently creating or editing—including the name and location of the base Windows Installer file, and any additional transforms that are associated with this transform and MSI file.

Options Dialog Box

The Options dialog box has two tabs: File Locations and View Settings.

File Locations

Within the File Locations tab, you can specify the default location of your source MSI files. This is reflected in the New Transform and Open Existing Transform Views of the Tuner Start Page.

View Settings

From the View Settings tab, you can select whether you want to display files from the base Windows Installer package (MSI) in the Files and Folders view in addition to files added in the transform.

Transform Summary Dialog Box

The Transform Summary dialog box, available from the Project menu, allows you to configure how to handle specific errors when the transform is applied. Additionally, you can configure how the Windows Installer Service verifies whether the transform can be applied against a given package.

Suppression Options

Options in this section allow you to configure whether installations with this transform applied will continue or fail if certain errors are encountered. You can configure the following options:

Table 12-8 • Suppression Options

Option	Description
Add Existing Row	Suppresses errors resulting from adding rows that already exist.
Delete Missing Row	Suppresses errors resulting from deleting rows that do not exist.
Add Existing Table	Suppresses errors resulting from adding existing tables.
Delete Missing Table	Suppresses errors resulting from deleting tables that do not exist.
Modify Missing Row	Suppresses errors resulting from updating rows that do not exist.
Change Code Page	Suppresses errors resulting from mismatched code pages.

Validation Options

Options in this section allow you to specify how the Windows Installer Service verifies the transform can be applied to a Windows Installer package. You can configure the following options:

Table 12-9 • Validation Options

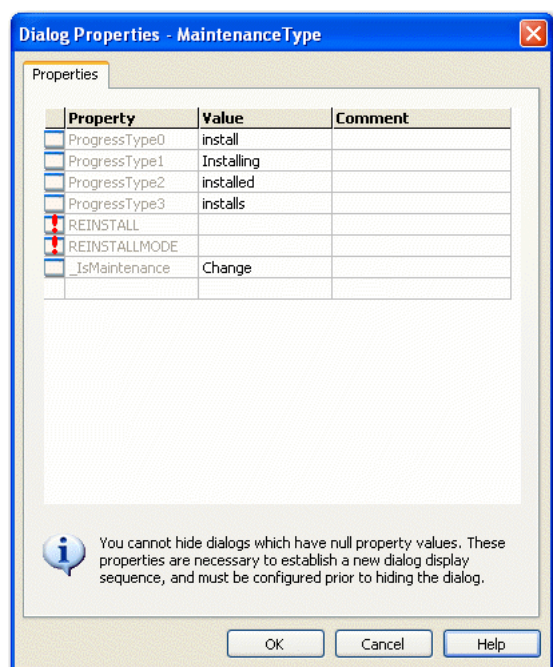
Option	Description
Same Language	If selected, the package against which the transform is applied must be the same language as the package used to create the transform.
Same Product Code	If selected, the product code for the package against which the transform is to be applied must be the same as the package product code of the package used to create the transform. If not selected, you can create a generic transform that can be applied against multiple Windows Installer packages.
Same Upgrade Code	If selected, the upgrade code of the package against which the transform is applied must be the same as the upgrade code of the package used to create the transform.
Product Version is Lower	If selected, the product version must be less than the version of the package used to create the transform. This can be combined with the Product Version is Equal option to create a “less than or equal to” comparison.
Product Version is Equal	If selected, the product version must be equal to the version of the package used to create the transform. This can also be combined with either the Product Version is Lower option or the Product Version is Higher option, creating a “less than or equal to” or “greater than or equal to” comparison.

Table 12-9 • Validation Options (cont.)

Option	Description
Product Version is Higher	If selected, the product version must be greater than the version of the package used to create the transform. This can be combined with the Product Version is Equal option to create a “greater than or equal to” comparison.
Version Checking	When using product version comparisons, you must indicate to what degree you want the comparison made. You can compare only the major versions, the major and minor versions, or the major, minor, and update versions. You can also select None to clear version checking.


Dialog Properties Dialog Box


You can use the Dialog Properties dialog box to view or change properties associated with UI sequence dialogs. The Dialog Properties dialog box is accessible by either selecting the appropriate dialog in the sequence and clicking Properties, or if you attempt to hide a dialog which has properties that must be configured.



Note • You must ascertain the purpose of each property from the Windows Installer package, as these properties are usually custom in nature. This is especially true for properties that must be set prior to hiding a dialog from the UI sequence.

Null Properties

You cannot hide UI sequence dialogs until you provide values for all currently null properties. These values are necessary to establish a new dialog box display sequence. Each property which must be configured prior to hiding the dialog in the sequence is marked with . Either configure the null properties, or click Cancel to return to the Dialogs view. Typically, configuring a null property involves clicking in the property's value field and selecting the value from the drop-down list.

Once you have provided a value for a null property, or if the property does not require configuration, it is denoted with .

Tuner Views

The following views are available in Tuner:

Table 12-10 • Tuner Views

Views	Subviews
Tuner Start Page	<ul style="list-style-type: none"> Welcome to Tuner View Create a New Transform View Open a Recent Transform View Open an Existing Transform View
Help View	
Package Validation View	<ul style="list-style-type: none"> Prevalidation View
Organization View	<ul style="list-style-type: none"> Product Properties View Features View
System Configuration View	<ul style="list-style-type: none"> Files and Folders View Registry View Shortcuts View INI Files View ODBC Resources View NT Services View
Application Configuration View	<ul style="list-style-type: none"> Server Locations View Setup Properties View Dialogs View Add/Remove Programs View
Package Preparation View	<ul style="list-style-type: none"> Postvalidation View Package View

Table 12-10 • Tuner Views (cont.)

Views	Subviews
Additional Tools View	<ul style="list-style-type: none">• Direct Editor

Tuner Start Page

From the Tuner Start Page, you can create a new transform project or open an existing one.

Select one of the following links for more information:

Table 12-11 • Tuner Start Page Subviews

View	Description
Welcome to Tuner View	General information about Tuner.
Create a New Transform View	Create a new customization, either starting with a blank transform, or by using a Response Transform based on selections from a custom installation.
Open a Recent Transform View	Select a transform previously created with Tuner.
Open an Existing Transform View	Select a transform created with a tool other than Tuner. You need to provide additional information, such as the base Windows Installer package for this transform.

Welcome to Tuner View

The Welcome view provides you with links to information on the Windows Installer service, Microsoft's integrated method of handling installations, as well as information about Tuner in the form of checklist steps. These steps cover the customization capabilities of Tuner.





Create a New Transform View

This view is displayed when you click **Create a new transform** or select **New** from the **File** menu.

This view contains the following options:

Base Windows Installer Package Subview

Table 12-12 • Base Windows Installer Package Subview

Field	Description
Select an MSI file	Enter the name and location of the Windows Installer package that you are customizing, or click Browse to locate it.
Provide a list of additional transforms to be applied	<p>If there are transforms already associated with the Windows Installer package, (for example, previous customizations or transforms containing language-specific information), click the New button ().</p> <p>When an entry appears in the list, click the Browse button (...) to the right of it and locate the transform.</p> <p>If multiple transforms are associated with this package, use the Move Up () and Move Down () buttons to specify the order in which the transforms are applied.</p> <p> Caution • When using multiple transforms, keep in mind that the order in which they are applied is critical. For example, if you create a transform for a Windows Installer package that creates a new value for a property, and then create a second transform that changes the value created in the first transform, everything works correctly. However, if you apply the second transform first, that transform is attempting to modify the property's value, instead of creating it. That will result in an error.</p> <p>One simple example of where this may be a problem is with the default company name. If the value is not set by default, and you set it in using the first transform, a new value for the property is created. If you create a second transform that modifies the combined original package and first transform, and the second transform changes the default company name, it is only changing the property. However, if you try to apply the second transform without the first one, Windows Installer interprets this as trying to change a null value to another value, which will result in an error.</p>

Windows Installer Transforms Subview

Table 12-13 • Windows Installer Transforms Subview

Field	Description
Provide a new project name and location (or accept the default) and click Create to create a new Customization project	<p>Provide a new project name and location. By default, the transform will be created in the same directory as the Windows Installer package, and named the same as the base package with an .mst extension.</p> <p>If you want to change the name and/or location of the transform, click Browse to open the Save Customization File dialog. Navigate to the directory in which you want to store the transform file you are creating. Provide the name of the transform with an .mst extension (for example, MyTransform.mst) and click Save. The dialog box closes and the path and file name appear in the edit field.</p>

Table 12-13 • Windows Installer Transforms Subview

Field	Description
Response Transform	<p>If you want to create a Response Transform, select this check box.</p> <p>If creating a Response Transform, step through the installation, making changes as necessary. When you reach the end of the installation sequence and click Install, the installation will exit and the Tuner interface will open your transform, which contains all of the changes you made during the simulated installation.</p>
Command line properties	<p>If you are using a response transform, you can specify additional command-line properties (in property name/value pairs separated by semicolons) to pass to the response transform. These must be PUBLIC properties, and only control how the dialogs are displayed during creation of the response transform. They are not persisted outside of the UI sequence during creation. For example, you can pass the property/value pair <code>ARPHELPTELEPHONE=1-111-111-1111</code> to set the value of the Help Telephone field of Add/Remove Programs.</p> <p>You might pass a property/value pair during response transform creation to display all dialogs during an installation that may not be displayed based on your system configuration (for example, to show Windows 9x-only dialogs on a Windows NT platform). You can then make appropriate responses and have them included in your transform.</p>



Note • You can access information about the original MSI file and associated transforms by selecting *Properties* from the *File* menu.

Open a Recent Transform View

This view, a list containing your most recently accessed transforms, is displayed when you click on Open a recent transform. Select a transform and click Open to open it, or select Properties from the context menu to view information about it (including details about the base MSI package and associated transforms).

You can also select one of the options at the bottom of the view to determine the view that is opened when you start up Tuner:

- load the last accessed transform when opening Tuner,
- make this recent list the default Tuner Start Page screen, or
- make the Welcome screen the default Tuner Start Page screen

Open an Existing Transform View

This view is displayed when you click on Open an existing transform or select File | Open. On this view, you can specify the name and location of the base Windows Installer package, any associated transforms, and the name and location of the transform file.

Generally, you will only use this option when opening existing transforms that were created by a product other than Tuner, or created by someone other than yourself. Transforms you create using Tuner are more easily accessed through using the Open a recent transform selection.

Help View

The Help view provides you instant access to this online help library. You can also access Microsoft's comprehensive Windows Installer reference library.

Package Validation View

The first recommended step in creating a transform is to perform validation on the base Windows Installer package. This helps you identify potential problems that you may or may not want to correct using Tuner.

Package validation is performed in the [Prevalidation View](#). To continue, click Prevalidation under Package Validation in the checklist.

Prevalidation View

The Prevalidation view provides you a way to ensure the base Windows Installer package for your transform is valid. If it fails the validation test, then unexpected (and unwanted) results can occur during installation.

To begin the prevalidation process, select the evaluation file that you want to use for package validation (or click Browse to locate it), and click Start. By default, the file is evaluated using the full logo-compliant validation file, and all [internal consistency evaluators](#) (ICEs) are checked. If you just want to test specific ICEs, after you select the evaluation file, specify the ICE names, and separate them by semicolons if there is more than one.

You can toggle the information level of the displayed results by checking the Show Info Messages, Show Error Messages, and Show Warning Messages check boxes. If any errors are present, the Windows Installer Package is invalid. Warning messages highlight potential problems, but will not cause validation to fail. Informational messages display ongoing information during the validation process.

Viewing the Prevalidation Results

As each ICE is run, Errors, Warnings, and Info messages are generated, and are listed in the Output tab at the bottom of the interface.

Upon completion of the Prevalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (❌), a Warning (⚠️), or an Informational Message (ℹ️), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row's associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red.

This feature makes it very easy for you to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform. For more information, see [Direct Editor](#).



Note • If no errors appear in the results (providing you are displaying errors), then the package is valid against the specific ICEs you specified, or against the entire evaluation file (if no ICEs were selected).

Organization View

The Organization view allows you to modify two main parts of the installation that your end users will see: the default path and default company name, and the actual features that can, will, or will not be installed.

Each subview of the Organization View is described below:

Table 12-14 • Organization View Subviews

Views	Description
Product Properties View	When a user runs a custom installation of a Windows Installer package, the Custom Setup dialog box provides a default installation path and a default organization name. The Product Properties view provides a mechanism for changing these defaults.
Features View	Features are the building blocks of the installation. They represent distinct pieces of functionality to end users, such as program files, help files, or clip art. You can modify which features and subfeatures are installed, and how they are installed, in the Features view.

Product Properties View

This view gives you a way to specify the default path on the user's computer into which the application will be installed. You can also specify the default organization name (i.e., your company's name).

The following properties are associated with this view:

Default Destination Variable

This is the name of the variable that stores the Default Destination Path. If you change this variable, you could create errors during postvalidation. Click on the variable's value to display a combo box that allows you to select a variable.

Generally, the variable used will be `INSTALLDIR` or `INSTALLLOCATION` (both author-created variables). However, another variable can appear as the value: `TARGETDIR` (a Windows Installer variable). If `TARGETDIR` is suggested, it is strongly recommended you contact the vendor who created the original MSI and ask what was used for the Default Destination Variable. While it is possible that it was `TARGETDIR`, it is also possible another variable was used and Tuner cannot identify this non-standard variable.

If the incorrect variable is set here, and/or if the Default Destination Path is changed, the installation may not function properly. If that happens, you can reset the information in this view by clicking on the Reset button when the Default Destination Variable is selected.

More information can be found in the [Windows Installer Help](#).

Default Destination Path

This location, stored within the Default Destination Variable, is the path where the application will be installed on the target machine, unless overridden during installation from the Custom Setup dialog. Click on the path's value to display a combo box of possible paths, or edit the path in the value field. It can be a hard-coded path, or it can be a Windows Installer folder property. Further levels can be separated with a backslash—for example, `ProgramFilesFolder\MyApp\Bin`.

To comply with Windows logo requirements, the application must default to a subfolder of `ProgramFilesFolder`, which can vary depending on the system's locale and user settings. If `ProgramFilesFolder\ProductName`, is specified as the default value for the Destination Folder property, then this product's files will always be installed to the logo-compliant location.

Company Name

This is the name the installation suggests for your organization during setup. If it is not set, the installer will automatically set it during installation using values from the registry. Once the value has been entered for the name of the organization, the `COMPANYNAME` property can be seen in the [Setup Properties View](#).

Features View

This view allows you to change Feature properties to best suit your situation.

Features are the building blocks of an installation from an end user's perspective. They represent a specific capability of the product, such as the help files or a part of a product suite that can be installed or uninstalled based on the end user's selections. Features can be composed of subfeatures, which in turn can be composed of further subfeatures. Depending on the visibility of the "parent" feature, end users can select which portions of a feature to install in the Custom Setup dialog.

Each feature and subfeature has properties that can be modified within Tuner. These include a description of each feature (as it appears in the Custom Setup dialog box), its visibility, and its initial state. Tuner allows you to change these feature properties to best suit your situation. For example, you may want to prevent a specific application within a suite from being installed in a particular transform file. By changing its initial state and visibility, you can prevent your end user from ever seeing this feature during installation.

The Features View contains the following options:

Table 12-15 • Features View Options

Options	Description
Description	This description will be displayed when this feature is clicked in the Custom Setup dialog box during installation.
Visible	<p>Specifies how the feature will be presented to the end user during installation in the Custom Setup dialog. The following options are available:</p> <ul style="list-style-type: none"> • Visible and Collapsed—This feature will be displayed in the Custom Setup dialog with its subfeatures collapsed by default. • Visible and Expanded—This feature will be displayed in the Custom Setup dialog with its subfeatures expanded by default. • Not Visible—This feature will not be displayed to the end user in the Custom Setup dialog. <p>Although an end user obviously cannot select or deselect an invisible feature, this property does not affect whether a feature is installed. In other words, a feature is not automatically installed if it is invisible; it just cannot be deselected if it would otherwise be installed, or selected if it should not be installed.</p>
Initial State	<p>Provide the initial state for the feature. The end user can override this from the Custom Setup dialog. Your options are:</p> <ul style="list-style-type: none"> • The feature is not installed—By default, the feature will not be installed during setup. • The feature is installed on the local drive—By default, the feature will be installed on the local drive during setup. • The feature is run from source, CD, or network—By default, the feature will be run from the source, whether it be from the installation CD or from the network. • The feature is advertised—By default, the feature will be advertised, but not installed. Essentially, this is an on-demand option; a shortcut will be created during setup, and if the shortcut is clicked, the feature will then be installed from the source. This ensures features that may be unnecessary are not installed until they are needed, if ever.

System Configuration View

The System Configuration view provides you with the ability to add additional files to a Windows Installer installation package, as well as add, remove, or modify shortcuts and registry information. Ultimately, this allows you to customize the installation to your needs, such as including company-specific templates in the correct folder during installation.

Files and Folders View

The Files and Folders View consists of four panes, representing the Source and Destination views.

Source Computer View

The Source view, located at the top, displays the folder and file structure on the user's computer.

Destination Computer View

The Destination computer's folders pane represents the folders on the target machine for the installation. The folders initially displayed for the target machine are ones used commonly in installations.

The Destination computer's files pane displays the files that are part of the installation. Initially, only files contained in the base Windows Installer package are displayed. When you add files to the package (into the transform), these also appear in this pane.



Note • Tuner cannot display files contained within compressed files.

A key file is a file that the Windows Installer uses to detect a component's presence. If the key file is in its proper location, the installer assumes that the entire component is installed correctly. Each component can have a key file, represented in the Files and Folders View by a key icon (🔑). The key files were set by the setup author, and cannot be modified using Tuner.

Destination Computer View Tasks

The following tasks are performed in the Destination Computer View:

- **Defining a New Folder**—To define a new folder, select either the Destination Computer or a predefined folder from the Destination Computer's Folders pane and select Add from the context menu.
- **Adding Files to an Installation**—To add files to an installation, simply locate them within the Source view, and either drag and drop them to the appropriate destination folder, or use the copy and paste commands.
- **Removing Folders or Files**—To remove folders and files you have already added to the installation, select the file or folder from the Destination view and select Delete from the context menu. Predefined folders are required for installation and cannot be removed. If you add files to an installation, they are always installed.
- **Removing Files from the Base Windows Installer Package**—Files from the base Windows Installer package can be removed during installation, except for key files. To remove a non-key file from the MSI, right-click on the file and select Remove from the context menu. The file is marked with an icon indicating it is not to be installed during installation.

Registry View

Similar to the Files and Folders View, the Registry view consists of four panes representing the Source and Destination views.

Source Computer View

The Source view, located at the top, displays the registry entries on the administrator's computer.

Destination Computer View

The Destination view represents the registry entries on the target machine for the installation. By default, the Destination view contains the following read-only registry hives: HKEY_CLASSES_ROOT, HKEY_CURRENT_USER, HKEY_LOCAL_MACHINE, HKEY_USERS, and HKEY_USER_SELECTABLE.

You can use the Registry view to:

- Create registry keys and values. You can do this using the same procedure you use to create registry keys and values in the Windows Registry Editor, or you can copy or drag and drop existing keys and values from the Source view to the Destination view.
- Import an existing REG file using the [Import REG File Wizard](#).
- Modify or delete registry keys that are part of the base installation. If you add new registry keys, they will always be installed.

Setting Registry Key Options

You must be cautious when creating new registry keys so adverse results do not occur. You must specify whether the key is to be created at install, removed at uninstall, or both. These mutually exclusive options are available by right-clicking on a key.

Table 12-16 • Registry Key Options

Option	Description
Create Key at Install	Creates the new registry key during installation if the key does not exist on the target machine.
Delete Key at Uninstall	Deletes the registry key during uninstallation, regardless of whether the key existed prior to the MSI's installation. This means that the key, and all its contents and sub-keys, will be removed regardless of whether other software information that is unrelated to this MSI exists. This can have a severe impact on other programs; only select this option if you are sure that the only software affected is the base MSI.
Both Create and Delete	Both of the above scenarios will occur.



Shortcuts View

The Shortcuts view offers an integrated, visual method for adding shortcuts and program folders to the installation. Existing shortcuts can also be modified or removed. For information on how to use the Shortcuts view to create, edit, or remove shortcuts, refer to one of the following topics:

- **Shortcut Properties**—Explains the Properties that you set when creating a shortcut.

- **Shortcut Locations**—Explains how to specify the location of a shortcut, either in a predefined folder or a folder that you create.
- **Shortcut Targets**—Explains how to specify the path to the shortcut's target application, batch file, folder, or feature.



Note • Shortcuts created in the transform are denoted by  and shortcuts from the base Windows Installer package are denoted by .

Shortcuts View/Shortcut Properties


When creating or editing a shortcut, you specify properties on the Shortcut Properties View.

The Shortcut Properties are described in the following table:

Table 12-17 • Shortcut Properties

Option	Description
Icon	<p>Displays the name of the icon for this shortcut.</p> <ul style="list-style-type: none"> • To change the icon, click the Change Icon button. On the Change Icon dialog box, click Browse to select an EXE file or DLL. All of the icons contained in that EXE or DLL file are displayed. Select the appropriate icon and click OK. On the Shortcut Properties view, the name of the icon now appears in the icon value cell and a picture of the icon appears next to the Change Icon button. • To remove the icon, click Clear Icon.
Description	A brief description of the shortcut. The text in this field will appear when users hover the mouse pointer over the shortcut in Windows 2000 or later.
Arguments	Any command-line arguments for the shortcut.
Target Type	<p>The destination folder, or a file from the MSI or transform. This property can be set to one of the following values:</p> <ul style="list-style-type: none"> • File from MSI Package—Used if the shortcut is to a file that is part of the base Windows Installer package. • File from System—Used if the shortcut is to a file that already exists on the target system. It may also be from the transform file. • Destination Folder—Used if the shortcut points to a folder rather than a file. The folder can be on the target system, from the MSI package, or from the transform. • Advertised shortcut—Used if you want this to be a shortcut to an “advertised” feature. Advertised features are not installed immediately during the setup process. Instead, they are installed when requested. The shortcut makes it appear that the feature is already installed, although it is not installed until the end user requests it.

Table 12-17 • Shortcut Properties (cont.)

Option	Description
Target	<p>Path and file name for this shortcut's target. There are many potential types of targets, including applications, files, folders, printers, and computers on a network. Instead of hard-coding a path, you can use a Windows Installer folder property in square brackets—for example, <code>[INSTALLDIR]\MyApp.exe</code>. You can also target Windows Installer features, which you can use for feature advertisement.</p> <p>To specify a feature as a shortcut target, enter the name of the feature in the Target field. You can determine the name of the feature by going to the MSI Table Editor and selecting the Feature table. The list of features you can target with your shortcut is listed in the Features column of the table.</p>
Run	<p>Specifies how the item is displayed when the shortcut is double-clicked. You can select from the following options:</p> <ul style="list-style-type: none"> • Normal Window—Launches the program in a normal sized window. • Maximized Window—Launches the program in full-screen view. • Minimized Window—Launches the program in a minimized window, visible only on the taskbar.
Working Directory	<p>Default directory for the Save As and Open dialogs. If you are modifying an existing shortcut, or creating a new one, you can select a Windows Installer folder property from the list instead of hard-coding a path. Separate further levels with a backslash—for example, <code><ProgramFilesFolders>\MyApp\Bin</code>.</p>
Hot Key	<p>The decimal value of the hot key combination for this shortcut. The Hot Key feature allows end users to launch a shortcut by pressing a combination of keys, rather than using the mouse. When you click in the Hot Key field to create or modify a shortcut, the Hot Key dialog box opens. While the Hot Key dialog box is open, press the desired hot key combination; those keystrokes are recorded. When you click OK on the Hot Key dialog box, Tuner automatically converts the keystrokes into a decimal value and enters that value in the Hot Key field.</p> <div>  <p>Caution • Microsoft recommends that you do not set this value, as it may conflict with existing hot key combinations on the target machine.</p> </div>

Shortcuts View/Shortcut Target

There are four shortcut Target Types you can add to a transform:

Table 12-18 • Shortcut Target Types

Target Type	Description
File from MSI Package	Used if the shortcut's target is a file that is part of the base Windows Installer package.
File from File System	Used if the shortcut's target is a file already existing on the target system.
Advertised Shortcut	Used if you want this to be a shortcut to an “advertised” feature. Advertised features are not installed immediately during the setup process. Instead, they are installed when requested. The shortcut makes it appear that the feature is already installed, although it is not installed until the end user requests it.
Destination Folder	Used if the shortcut points to a folder rather than a file.

The Target Type that is selected affects what you should enter in the Target property field:

File from MSI Package & File from File System

Under most circumstances, shortcut targets are applications or batch files. You simply provide the full path to the application or batch file in the Target property. However, after entering the target, if you leave the Shortcuts view and then return to it, you find that the target has changed. For example, you might have entered `C:\Temp\MyFolder\mytarget.exe` as the target, but it now reads `[MyFolder]mytarget.exe`. What has happened is that the path has been replaced based on entries made to the Directory table. For more information, see [Determining the Path of Changed Shortcuts](#).

By stringing together the directories you just located in the Directory table, you can determine the path represented by `[MyFolder]` in the shortcut target. If you use the drop-down list in the Target property, you can determine the absolute values of these other directories in the same fashion.

Destination Folder

To point the shortcut to a folder rather than a file, select Destination Folder in the Target Type property, and then select a folder name from the Target property drop-down list. The Target list includes available folders on the target system, from the MSI package, and from the transform.

Advertised Shortcut

You can also target Windows Installer features, which you can use for feature advertisement. To specify a feature as a shortcut target, simply enter the feature name in the Target field. You can determine the name of the feature by going to the Direct Editor and selecting the Feature table. The list of features that you can target is listed in the Features column of the table.

Shortcuts View/Shortcut Locations

When you first navigate to the Shortcuts view, you see a set of predefined folders with existing shortcuts (if the base MSI package had shortcuts defined). You can modify or remove these shortcuts according to your needs.

If you need to create your own shortcut, you can place it in a new folder you define or in a predefined folder. Additional predefined folders that are not displayed can be accessed by right-clicking on the uppermost item in the Shortcuts explorer and selecting Show Folder.

Under the Show Folder submenu is a list of the additional predefined folders supported in the Shortcuts view. Select the folder where you want your shortcut created to have it displayed in the Shortcuts explorer. The predefined shortcut destinations are described below.

Alternately, you can create your own folder in which to place shortcuts by right-clicking either a folder or the top level Shortcuts item and selecting New Folder. To remove a folder that you have added, right-click on it and select Delete.

The following predefined folders are available for shortcuts:

Table 12-19 • Predefined Shortcut Folders

Predefined Folder Name	Description
AppDataFolder	The current user's Application Data folder.
CommonFilesFolder	The Common Files folder for the current user.
DesktopFolder	The user's desktop. Although placing a shortcut on the desktop makes it easily visible, it can also be distracting to users, so it should be used sparingly.
FavoritesFolder	he Favorites folder for the current user.
FontsFolder	References the target machine's Fonts folder.
INSTALLDIR	The installation's default destination folder.
ProgramFilesFolder	References the target machine's Program Files folder.
ProgramMenuFolder	The Program menu for the current user.
SendToFolder	The user's Send To folder, which is accessible when you right-click on files. Shortcuts are placed here so users can have quick access to the target program from many file types.
StartMenuFolder	The Start menu folder for the current user.
StartupFolder	The current user's Startup folder. Shortcuts placed here automatically launch their targets whenever Windows is started.
SystemFolder	The target machine's System folder.



Table 12-19 • Predefined Shortcut Folders (cont.)

Predefined Folder Name	Description
TempFolder	References the target machine's Temp folder (usually C:\Temp).
TemplateFolder	The current user's Template folder.
WindowsFolder	The target machine's Windows folder.

INI Files View

Initialization (INI) files serve as a database in which you can store and retrieve information between uses of your application. Typically, INI Files contain key name-value pairs representing run-time options for applications. Some INI files, such as Boot.ini and Wininit.ini, are used by the operating system.

The INI Files view provides a graphical way for users to add, modify, or delete the contents of the IniFile Table.

- The INI Files view displays the contents of the IniFile table from the source Windows Installer package  and the transform .
- The view itself consists of three panes: the leftmost a tree of predefined folders from the Windows Installer package and user-defined folders from the transform.
- The top-right section displays the keys and values in the selected IniFile section. Windows Installer and transform values are distinguished by different icons in this pane.
- The lower-right pane provides information about the selected key.

Editing INI Files in Tuner

To edit an INI File in Tuner, simply expand the appropriate IniFile node in the tree. Then select the appropriate section, which appears in the upper right pane. You can then edit the keys and values appropriately. You can also insert new keys and values by right-clicking in the key and value pane and selecting Add. If you want to add a new section to an INI File, right-click on the INI File in the tree and select New Section. You can also delete an INI File, a section of an INI File, or a key by right-clicking the appropriate node or property sheet entry and selecting Remove.

For detailed instructions on performing these tasks, click on one of the following topics:

- [Adding INI Files](#)
- [Adding New Keys to INI File Sections](#)
- [Adding Sections to INI Files](#)
- [Importing Existing INI Files](#)
- [Modifying INI File Keys, Values, and Actions](#)
- [Removing INI Files](#)
- [Removing INI File Section Keys](#)

- [Removing Sections from INI Files](#)
- [System Configuration View](#)

ODBC Resources View

Open Database Connectivity (ODBC) Resources are ones that involve interaction with databases. Tuner allows you to view existing ODBC Data Sources, ODBC Drivers, and ODBC Translators.

The left pane of the ODBC Resources view contains a tree with the three root nodes: ODBC Data Sources, ODBC Drivers, and ODBC Translators. When any of these are expanded, individual Data Sources, Drivers, and Translators contained in the Windows Installer package are displayed. When selected, each of these individual nodes displays information in a property grid displayed in the upper right pane.

There are three different types of ODBC Resources available for viewing and/or modification through the ODBC Resources view:

Table 12-20 • ODBC Resource Types

Resource Type	Description
ODBC Data Sources	<p>The source of the data (database type) and information on how to connect to that database. Common database types include Microsoft SQL Server, Microsoft Access, and Visual FoxPro. Connection information may include the name of the database, where the server that hosts it is located, and logon/password information. You can add new ODBC Data Sources from the ones existing on your computer, or delete ones you add or existing ones from the MSI. You can also add, edit, and delete ODBC Data Source attributes. If your machine does not have the ODBC Data Source that is needed by the package, you can type it into the ODBC Data Source dialog.</p> <p>See ODBC Resources and Adding New Data Sources for more information.</p>
ODBC Drivers	<p>These are libraries that implement functions involving ODBC. Each database type has its own ODBC driver. You can add only those Data Sources for which ODBC Drivers exist in the MSI package. You can add, edit, or delete new attributes for ODBC Drivers, and you can edit or delete all attributes except for File, Setup File, and Feature.</p> <p>See Adding New ODBC Driver Attributes and Editing ODBC Driver Attributes for more information.</p>
ODBC Translators	<p>These translate one form of raw data into another form that can be used with a specific database type. For example, an ODBC translator may convert from one code package to another. You can only view the contents of an ODBC Translator and cannot add, delete or modify them.</p>



Note • Only ODBC Data Source attributes are editable; ODBC Drivers and ODBC Translators are provided in read-only form.

NT Services View

The NT Services view provides a way to change parameters for NT Services included in the base Windows Installer package.



Note • NT services cannot be added to a setup using Tuner. You can only make modifications to services in the base Windows Installer package.

The following options can be modified:

Table 12-21 • NT Services View Options

Option	Description
Name	This property contains the name of the service to install. This property may have the same value as the Display Name, but is used by the installer in a different way.
Display Name	The name of the service as it appears in user interfaces (such as the name used under the NT Services control panel). This string can be a maximum of 256 characters in length. It may be the same as the Name property.
Service Type	There are two service types available: <ul style="list-style-type: none"> • Service that runs in its own process • Service that shares a process with others
Interact with Desktop	Although uncommon, some services need to interact with the desktop to display message or dialog boxes for the user. If this service requires this functionality, this property's value is set to Yes.

Table 12-21 • NT Services View Options (cont.)

Option	Description
Start Type	<p>The value in this property dictates when the service is started. The possible values are:</p> <ul style="list-style-type: none"> • Automatic—The service starts during system startup. • Manual—The service starts when the service control manager calls the StartService function. • Disabled—The service cannot be started. <p>There are two additional values, available only for driver services:</p> <ul style="list-style-type: none"> • Start at boot time—The device driver is started by the operating system loader. • Started by the system—The device driver is started by calling the IoInitSystem function.
Error Control	<p>This property specifies what action is taken by the startup program should the service fail to start properly during startup. The available values are:</p> <ul style="list-style-type: none"> • Ignore Error—Logs the error and continues startup. • Normal Error—Logs the error, displays a message box informing the user of the problem, and continues startup. • Critical Error—Logs the error, if possible, and restarts the system with the last-known-good configuration. If the last-known-good configuration caused the failure, the startup operation fails.
Overall Install	<p>This property's value specifies how the installation handles a situation when this service cannot be installed for some reason. There are two possible resolutions:</p> <ul style="list-style-type: none"> • Continue overall install if service fails to install • Fail overall install if service fails to install
Load Order Group	<p>The value of this property is a string that names the load ordering group of which this service is a member. If the service does not belong to a load order group, this value should be either an empty string or NULL.</p>
Dependencies	<p>A list of names of services or load ordering groups that the system must start prior to starting this service.</p>
Start Name	<p>The name under which the service is logged on. Leaving this field blank causes the service to be installed for the LocalSystem account.</p> <p>See the Windows Installer help topic ServiceInstall Table for information on the format for the StartName value.</p>
Password	<p>The password associated with the start name. Most services will have a blank value for this property.</p>

Table 12-21 • NT Services View Options (cont.)

Option	Description
Arguments	This property contains any command-line arguments or properties required to run the service.
Description	This property contains a localizable description of the service. It is typically set by the setup author.
Feature	This read-only property contains the name of the feature with which this service is associated.

Application Configuration View

Application Configuration in Tuner involves adding or modifying properties that affect your setup as well as specifying properties for Add/Remove Programs in Control Panel for Windows 2000 and XP. You can also configure source resiliency using the Server Locations view, and customize user interface sequences from the Dialogs view.

The Windows 2000 and Add/Remove Programs in Control Panel differs from the previous Windows operating systems in many ways. Depending on how the Windows Installer setup is configured, the user has the option of removing, repairing, or changing the installation with the click of a button. Windows 2000 users are also be able to access additional information in Add/Remove Programs not available on previous platforms. With this information, it is easier for your end users to find technical support links, phone numbers, product update information, and information about your company.

Add/Remove Programs functionality can also be disabled to limit the number of end users who have access this feature.

Server Locations View

If you install from a network server, and if you install features to run from the server or that will be advertised for installation on their first use, the applications may need access to the server sometime after the initial installation. The applications may also require access to the server if a file is deleted or becomes corrupt, as the application can copy the problematic file(s) automatically from the server.

To ensure that users always have access to an available network server for these circumstances, you can copy the administrative installation to one or more additional servers, and then specify those servers from within this view. If the primary server should become unavailable, the Windows Installer will attempt to connect to the other servers specified here, in the order they are specified. If no server is found, the Windows Installer will prompt the user to specify the location of the server.

Setup Properties View

Even though Tuner provides you views to customize many areas of the Windows Installer package, it may be necessary to edit property values that are not available elsewhere. The Setup Properties view exposes the entries in the properties table (the underlying structure of Windows Installer packages). You can also add your own custom properties here.

Properties exist in two formats: Private and Public. Private properties are set by the software vendor or by the Windows Installer during installation and cannot be altered. Private properties are always lowercase, and appear in Tuner in grayed out text. Public properties, which are always in capital letters, can also be set by the software vendor, but can be edited. Tuner also allows the addition of Public properties to the transform. The Public properties that you create can be edited or removed as necessary, whereas preexisting Public properties can only be edited.



Caution • Before you begin changing properties in the Setup Properties view, ensure you know exactly what you are doing. The changes you make may cause validation errors, installation errors, or other unforeseen problems.

Dialogs View

When customizing the Windows Installer package, you may want to disable particular panels that appear during the installation, administrative, patch, or maintenance sequences. You can do so from the Dialogs view.

This view contains a list of each of the four installation modes (installation, administrative, maintenance, and patch), with the associated dialogs that appear as part of the UI sequence during the selected mode. You can enable or disable these dialogs by either the check box to the left of the dialog name, or by using the Show and Hide buttons.

If you hide a dialog that appears in more than one sequence, the dialog is hidden during all sequences.



Note • During each installation mode, Windows Installer displays a Wizard containing a sequence of panels. However, the underlying Windows Installer technology actually uses a series of dialogs displayed in sequence. During runtime, they are referred to as panels (as with other Wizards); at design time, they are individual dialogs that can be enabled or disabled as necessary.

Add/Remove Programs View

The Windows 2000 and Windows XP Add/Remove Programs in Control Panel differs from the previous Windows operating systems in many ways. Depending on how the Windows Installer setup is configured, the user has the option of removing, repairing, or changing the installation with the click of a button.

Windows 2000 and XP users are also able to access additional information in Add/Remove Programs in Control Panel not available on previous platforms. With this information, it is easier for your end users to find technical support links and phone numbers, product update information, and information about your company.

You can set the following options from the Add/Remove Programs view:

Table 12-22 • Add/Remove Programs View Options

Option	Description
Publisher URL	Contains a URL for the publisher's home page. Corresponds to the ARPURLINFOABOUT property in the Setup Properties view.
Product Info and Update URL	Contains a URL that links to update information for the application. Corresponds to the ARPURLUPDATEINFO property in the Setup Properties view.
Help URL	Contains the Internet address for technical support. Product maintenance applets display this value. Corresponds to the ARPHELPLINK property in the Setup Properties view.
Help Telephone	Contains the telephone number that users can call for assistance with the product. Corresponds to the ARPHELPTELEPHONE property in the Setup Properties view.
Contact Person	Contains a the name of the person to contact for help or information about the product. Corresponds to the ARPCONTACT property in the Setup Properties view.
Comments	Contains additional information that is provided for the user. Corresponds to the ARPCOMMENTS property in the Setup Properties view.
Disable Modify Button	Provides a way to prevent users from running the application setup to modify the product's installation. Corresponds to the ARPNOMODIFY property in the Setup Properties view.
Disable Remove Button	Provides a way to prevent users from running the application setup to remove (uninstall) the product from the user's computer. Corresponds to the ARPNOREMOVE property in the Setup Properties view.
Disable Repair Button	Provides a way to prevent users from running the application setup to repair missing or corrupt product files. Corresponds to the ARPNOREPAIR property in the Setup Properties view.

Package Preparation View

The final step in creating a customization involves two parts. First, you should postvalidate your transform and base Windows Installer package. This ensures that you have not introduced any errors into the installation, and may help you verify that you have corrected errors that existed in the base package. Secondly, you need to actually package the transform and base package for distribution.

These steps are carried out in the following Package Preparation View subviews:

Table 12-23 • Package Preparation View Subviews

Views	Description
Postvalidation View	Allows you to ensure that your Windows Installer package is valid. The difference is that it also checks your newly created transform to make sure it is valid in relation to the base package.
Package View	When you have finished with your transform and postvalidation, the last step is to prepare the overall package so you can distribute it to your users (using a third-party tool such as Microsoft SMS).

Postvalidation View

Much like the Prevalidation view, the Postvalidation View allows you to ensure that your Windows Installer package is valid. The difference is that it also checks your newly created transform to ensure it is valid in relation to the base package. You can run the same internal consistency evaluators in the evaluation file, and receive the report back on the overall package and transform validity. By default, all [ICEs](#) are checked for the specified evaluation file.

You can select the information level of the displayed results by checking the Show Info Messages, Show Error Messages, and Show Warning Messages check boxes. If any errors are present, the Windows Installer package is invalid. Warning messages highlight potential problems, but will not cause validation to fail. Informational messages display ongoing information during the validation process.

If you started off with a valid Windows Installer package, yet postvalidation fails, it is likely your problems relate to changes you made in the transform. Make sure you look at the [Evaluation Files and Internal Consistency Evaluators](#) topic to see what each ICE message means. You can also consult the online MSI Help reference, available from the Help Menu, for more details.

It is also possible to start off with an invalid base package, but the postvalidation does not have any errors. If this happens, the properties you changed in your transform can bring the overall package and transform up to a valid package.

Output of the postvalidation appears in the Output and Validation tabs of the Output Window.

Viewing the Postvalidation Results

As each ICE is run, Errors, Warnings, and Info messages are generated, and are listed in the Output tab at the bottom of the interface.

Upon completion of the Postvalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (❌), a Warning (⚠️), or an Informational Message (ℹ️), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row's associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red.

This feature makes it very easy for you to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform. For more information, see [Direct Editor](#).



Note • If no errors appear in the results (providing you are displaying errors), then the package and transform are valid against the specific ICEs you specified, or against the entire evaluation file (if no ICEs were selected).



Tip • It is possible for a package that passed the prevalidation to fail the postvalidation. Remember changes made in the Setup Properties can affect your installation. If your package fails postvalidation, check all changes made in the Setup Properties for accuracy. To identify the original Setup Properties, you can create a new transform file that can be deleted at any time. Changes made using the Direct Editor can also affect your installation's functionality.

Package View

When you have finished with your transform and postvalidation, the last step is to prepare the overall package so you can distribute it to your users (using a third-party tool such as Microsoft SMS). Tuner provides several different packaging options that can be used individually or in conjunction with one another. Note that Tuner does not actually perform the distribution; rather, it gathers the necessary files together, and copies them to a location you specify. You can then use some of the standard software distribution tools to roll out your customized package.

The three subviews contained within the Package view are:

Table 12-24 • Package View Subviews

View	Description
Package View/Location View	Place the transform and base Windows Installer file on either an FTP server or network location.
Package View/Setup View	Package the transform and base Windows Installer file with an executable launcher (Setup.exe) to begin the installation. You have the option of including the MSI Engine for the appropriate platform to ensure Windows Installer functionality. Setup.exe uses information contained in setup.ini to determine the package, associated transform, and any command-line parameters.
Package View/SMS View	Package and prepare the transform and Windows Installer file for distribution using Microsoft SMS. Tuner can create a PDF file and/or an SMS file for your package and transform.



Note • These three subviews are also available through the Packaging Wizard, which can be accessed from the Project menu by selecting Package.

Package View/Location View

It is from the Location View that you specify information regarding where to place the transform and initial Windows Installer file. You have the option to copy files to a network location, including performing an administrative installation, or copy files to an FTP server.

Package View/Setup View

In the Setup view, you can set options that are used to create a Setup.exe file to launch both your transform and Windows Installer package. You have the option to include the MSI engine for Windows 95/98 and NT, as well as include any command-line arguments for the installation.

Package View/SMS View

In many cases, you may want to perform distribution of the transform and Windows Installer package using Microsoft SMS. Tuner provides a way to create both a Package Definition File (PDF) or an SMS file in the SMS View.

Select the file type(s) you want to create. If you create an SMS file, Tuner can instruct SMS to create a Management Information Format (MIF) file when SMS deploys the package and transform. If you want to do this, provide the Install MIF Filename, Uninstall MIF Filename, and serial number.

For more information about Microsoft SMS, consult the SMS documentation.

Additional Tools View

Tuner includes an extremely flexible additional tool: the Direct Editor. Using this tool, you can directly edit the Windows Installer tables that make up the Windows Installer package. This provides you with extremely granular control over the transform you are creating.

Direct Editor

Windows Installer packages are relational databases consisting of dozens of interrelated tables. These tables reflect the application's features, components, relationship between features and components, registry information, and user interface.

The Direct Editor allows you to edit values in the MSI tables of the base Windows Installer package and store them in your transform. As you change values elsewhere in your transform, those changes are reflected in the Direct Editor, and vice versa. The complete list of MSI tables contained in the installation package is displayed in the left pane. When you select a table, the contents are displayed in the right pane.

Working Directly with MSI Tables

The Tuner Direct Editor provides the ability to work directly with MSI tables. This includes the ability to edit the contents, as well as find and replace values.



Tip • When viewing or editing specific tables, pressing F1 launches the Microsoft Windows Installer help system to the appropriate table, if it is a standard Windows Installer table. When F1 is pressed while viewing a non-standard table, the Windows Installer help system launches to its default topic. Consult the software vendor for information about custom tables.

Table Functionality

The following functionality is available for tables:

Table 12-25 • Direct Editor Table Functionality

Function	Keyboard Shortcut	Description
Add Records	Insert	Adds a new record to the table.
Delete Records	Del	Deletes the selected record after user confirmation. Referential integrity is not maintained.
Cut Row(s)	Ctrl+X	Enables users to cut single or multiple rows or cells in the grid to the clipboard.
Copy Row(s)	Ctrl+C	Copies the selected cell or row in the grid to the clipboard.
Paste Row(s)	Ctrl+V	Pastes the contents of the clipboard into a given cell or row(s).

Editing Tables by Launching the Direct Editor from the Validation Tab

Upon completion of a Pre- or Postvalidation, the Validation tab is automatically selected, and all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (❌), a Warning (⚠️), or an Informational Message (ℹ️), the name of the ICE that generated it, and a brief description of what caused it to occur.

If a row is grayed out, it indicates that the table cannot be edited in the Direct Editor (perhaps because it is in an external package). If a row is active, you can double-click on it to open that row's associated table. The Direct Editor is launched and the table and/or table cells that are causing the problem are highlighted in red.

This feature makes it very easy for you to use the Direct Editor to edit values in the MSI tables of the base Windows Installer package and store them in your transform.

Resizing Table Columns in the Direct Editor

When you initially open the Direct Editor, the selected table's columns are listed in a compact format so that the maximum number of columns are displayed.

To automatically resize a column so that its width matches that of its longest entry, double-click on the column heading. This new column width setting is automatically saved and will be implemented the next time you view this table column in the Direct Editor.

Sorting Table Columns in the Direct Editor

To sort a table column, click the column heading once. The order will toggle between ascending and descending.

Import INI File Wizard

Tuner allows you to import any existing INI files that you may have created previously. To import a INI file, you need to launch the Import INI File Wizard.

The Wizard consists of the following panels:

- [Welcome Panel](#)
- [Import INI File Panel](#)
- [Import Conflict Options Panel](#)
- [Finishing INI File Import Panel](#)

Within the [INI Files View](#), right-click on a folder under the Destination Computer node (or right-click on the Destination Computer node to add a folder first), and then select Import INI File. The Wizard that appears prompts you for the location of the INI file, as well as what to do when there are conflicts arising from duplicate values. When import occurs, Tuner merges the contents of the INI file with existing INI file data.

Welcome Panel

The Import INI File Wizard allows you to import data contained in an INI file into your transform.

Import INI File Panel

From this panel, you need to specify the name of the INI file (.ini) you want to import into your transform. Alternatively, click Browse and navigate to it.

Import Conflict Options Panel

The Import Conflict Options panel allows you to specify how you want to handle duplicate INI file data.

Select one of the following options for the Wizard to use to determine how to handle these conflicts:

Table 12-26 • Import Conflict Options

Option	Description
Overwrite the data in the IniFile table	If conflicts exist, the Wizard will overwrite the INI file keys and values with any duplicate keys from the registry file (.reg).
Do not overwrite the data in the IniFile table	If duplicate keys and values are found, the Wizard will retain the existing INI file data and not overwrite it.

Click Import to import the .ini file. When the file has been imported, the [Finishing INI File Import Panel](#) is displayed.

Finishing INI File Import Panel

This panel appears following import of the .ini file. Click Finish to exit the Wizard and return to the [INI Files View](#).

Import REG File Wizard

Tuner allows you to import any existing REG files that you may have created previously. To import a REG file, you need to launch the Import REG File Wizard.

The Wizard consists of the following panels:

- [Welcome Panel](#)
- [Import Registry File Panel](#)
- [Import Conflict Options Panel](#)
- [Finishing Registry Import Panel](#)

To launch the Import REG File Wizard, go to the [Registry View](#), right-click on one of the registry hives or on a registry key you have added, and select Import REG File from the context menu.

The Wizard that appears prompting you for the location of the registry file, as well as what to do when there are conflicts arising from duplicate keys. When import occurs, Tuner merges the contents of the REG file with existing registry data.

Welcome Panel

The Import REG File Wizard allows you to add registry data contained in a registry file (.reg) into your transform.

Import Registry File Panel

From this panel, you need to specify the name of the registry file (.reg) you want to import into your transform. Alternately, click Browse and navigate to it.

Import Conflict Options Panel

The Import Conflict Options panel allows you to specify how you want to handle duplicate registry keys and values.

Select one of the following options for the Wizard to use to determine how to handle these conflicts:

Table 12-27 • Import Conflict Options


Option	Description
Overwrite the registry data	If conflicts exist, the Wizard will overwrite the registry keys and values with any duplicate keys from the registry file (.reg).
Do not overwrite the registry data	If duplicate keys and values are found, the Wizard will retain the existing registry data and not overwrite it.

Click Import to import the .reg file. When the file has been imported, the [Finishing Registry Import Panel](#) opens.

Finishing Registry Import Panel

This panel appears following import of the .reg file. Click Finish to exit the Wizard and return to the [Registry View](#).

Packaging Wizard

The Packaging Wizard provides a way to step through the packaging process for the transform and Windows Installer package. Please note that this packaging merely places the installation on a network location or FTP server, creates a Setup.exe file, and/or creates files for SMS distribution. It does not actually distribute the installation to client machines. To invoke the Packaging Wizard, select Packaging Wizard from the Project menu, or select the Packaging Wizard button () from the toolbar.

The Packaging Wizard consists of the following four panels:

- [Location Panel](#)
- [Setup.exe Panel](#)
- [SMS Panel](#)
- [Packaging Summary Panel](#)

Location Panel

The first panel of the Packaging Wizard allows you to specify the location to store the installation files (including transforms). If you select Network Location, you can specify or browse to the directory location.

Alternately, you can copy the installation files to an FTP server. If you select this option, you must specify the FTP location (URL), the user name under which to log in, and the password.

Setup.exe Panel

This Packaging Wizard panel allows you to include a setup.exe launcher for your package and transform, and include the appropriate MSI engine for Windows 9x or NT. You can also specify command-line arguments for the Windows Installer.

SMS Panel

In many cases, you may want to perform distribution of the transform and Windows Installer package using Microsoft SMS. Tuner provides a way to create both a Package Definition File (PDF) or an SMS file in the SMS View. Select the file type(s) you want to create. If you create an SMS file, Tuner can instruct SMS to create a Management Information Format (MIF) file when SMS deploys the package and transform. If you want to do this, provide the Install MIF Filename, Uninstall MIF Filename, and serial number.

Packaging Summary Panel

The Package Summary panel informs you of the packaging options selected in the three previous panels. If you need to make changes, use the Back button to return to the previous panel. The Cancel button aborts the packaging operation. If you are satisfied with the selected options, click Finish to copy the installation files to the specified location and/or create Setup.exe and SMS files.

Part 4

Testing Windows Installer Packages

This part of the AdminStudio 9.5 User Guide includes the following chapters:

- [Identifying and Resolving Application Conflicts Using ConflictSolver](#)
- [Identifying and Resolving Package Errors Using PackageExpert](#)
- [Analyzing the Impact of Installing Microsoft Operating System Patches](#)
- [Isolating Applications Using Application Isolation Wizard](#)
- [Ensuring Package Quality Using QualityMonitor](#)

Identifying and Resolving Application Conflicts Using ConflictSolver



Edition • ConflictSolver is included with AdminStudio Professional and Enterprise Editions.

Although a Windows Installer package or merge module may be built to guidelines put forth by Microsoft, it is possible that the interaction between packages, or between a package and the base operating system, may cause unwanted results in your production environment. ConflictSolver enables you to identify these conflicts before you deploy packages, and resolve the problems before they affect your end users.

ConflictSolver allows you to check for a variety of conflict types, including components, files, registry entries, shortcuts, INI files, ODBC resources, NT services, file extensions, product codes, and merge module conflicts. In many cases, ConflictSolver can resolve the issues automatically. You can also create your own custom rules to ensure packages conform to your internal standards and practices.

The ConflictSolver user documentation is organized into the following areas:

Table 13-1 • ConflictSolver User Documentation

Section	Description
About ConflictSolver	Compares ConflictSolver and Application Manager, describes ConflictSolver views, explains how to connect to Application Catalogs from ConflictSolver, and describes the difference between Conflict ACE Rules and Best Practice ACE Rules.
Best Practices for Conflict Detection	Explains the “best practices” you should take to ensure optimal performance of ConflictSolver during package import and conflict detection.
Validating Packages	Optimally, packages should pass ICE validation. Topics in this section instruct you how to perform validation, and what to do if it fails.
Identifying and Resolving Conflicts	ConflictSolver’s core purpose is to help you identify and resolve application conflicts. Topics in this section show you how.

Table 13-1 • ConflictSolver User Documentation (cont.)

Section	Description
Generating Reports	You can generate both pre-defined and user-defined reports using ConflictSolver.
Application Conflict Evaluators (ACEs)	ConflictSolver includes many rules that are used for identifying package conflicts. This section covers each rule, and also discusses how to create your own rules.
Conflict Application Resolution Definitions (CARDs)	Conflict Application Resolution Definitions (CARDs) are the rules used to fix conflicts identified by Application Conflict Evaluators (ACEs) when conflict detection is performed.
ConflictSolver Reference	Every view, dialog box, and Wizard available in ConflictSolver is discussed in this section.

About ConflictSolver

This section includes the following topics:

- [Application Manager and ConflictSolver](#)
- [Types of ACE Rules: Best Practice and Conflict](#)
- [Connecting to Existing Application Catalogs](#)

Application Manager and ConflictSolver

In previous releases, ConflictSolver was the only tool that provided a comprehensive view of the Application Catalog. Consequently, ConflictSolver was used to perform functions that are not related to conflict detection or resolution, such as importing and storing packages in a database, and organizing those packages.


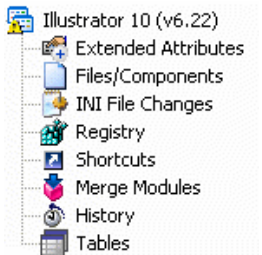

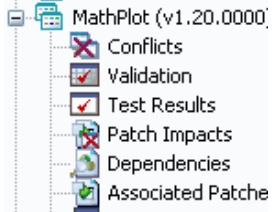
To reduce the complexity of ConflictSolver, the functionality that was being provided by ConflictSolver has been divided into two tools: ConflictSolver and Application Manager. This section compares these tools and includes the following topics:

- [Functionality Comparison](#)
- [Accessing the Tools](#)
- [Displaying ConflictSolver Options in Application Manager](#)
- [Creating and Connecting to Application Catalogs](#)

Functionality Comparison

This table compares the functionality included in Application Manager and ConflictSolver.

Table 13-2 • Comparison of ConflictSolver and Application Manager Functionality

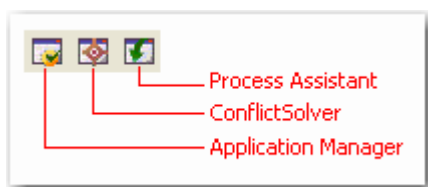
View	Major Functionality	Package Sub-Nodes
Application Manager 	<p>Use to perform the following tasks</p> <ul style="list-style-type: none"> • Import packages—Import packages into the Application Catalog. • Manage packages—Organize packages into Groups, rename packages, and delete packages. • Merge Wizard—Merge data from a source Application Catalog into the currently open Application Catalog. • Package Auto Import—Set up the automatic import of packages. • Replication—Makes it possible to distribute an Application Catalog database where it is needed across a large enterprise. • Software Repository—Import packages into and check in and check out packages from the Software Repository. • Associate a package with a Workflow Manager package • Launch applications—Launch Predeployment Test Preparation Wizard, InstallShield Editor, and Distribution Wizard. • Reports—View Package, File, and Registry reports. 	 <ul style="list-style-type: none"> Illustrator 10 (v6.22) <ul style="list-style-type: none"> Extended Attributes Files/Components INI File Changes Registry Shortcuts Merge Modules History Tables
ConflictSolver 	<p>Use ConflictSolver to perform the following tasks:</p> <ul style="list-style-type: none"> • Conflict Analysis and Resolution—Check one or more packages for conflicts against others in the Application Catalog, and resolve those conflicts. • Validation—Validate Windows Installer package against Microsoft ICEs (Internal Consistency Evaluators), • Dependency Scan—Generate a list of all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog. • Patch Impact Analysis Wizard—Launch Patch Impact Analysis Wizard for selected package. • Reports—View Package, File, and Registry reports. <p>For information on using ConflictSolver, see Identifying and Resolving Application Conflicts Using ConflictSolver.</p>	 <ul style="list-style-type: none"> MathPlot (v1.20.0000) <ul style="list-style-type: none"> Conflicts Validation Test Results Patch Impacts Dependencies Associated Patche

Accessing the Tools

Both Application Manager and ConflictSolver can be opened from the Start Page, the Tools Gallery, or from a menu shortcut.



You can toggle between Application Manager, ConflictSolver, and ConflictSolver Process Assistant by clicking a button in the toolbar:

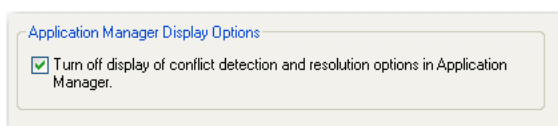


When the ConflictSolver or Application Manager is active, **ConflictSolver** or **Application Manager** is listed in the Title bar and also in the top right corner of the interface.

Displaying ConflictSolver Options in Application Manager

By default, all functionality for interacting with the Application Catalog, including conflict detection and resolution, is available in Application Manager.

To turn off the display of conflict detection and resolution options in Application Manager, select the checkbox in the **Application Manager Display Options** area of the **General** tab of the ConflictSolver and Application Manager [Options Dialog Box](#).



Creating and Connecting to Application Catalogs

You can connect to Application Catalogs from both Application Manager and ConflictSolver as described in [Connecting to an Existing Application Catalog](#). However, you use the AdminStudio interface to create new application catalogs. See [Creating New Application Catalogs](#).

Types of ACE Rules: Best Practice and Conflict

When performing conflict analysis with ConflictSolver, you will be using two different types of ACE Rules:

- [Best Practice ACE Rules](#)
- [Conflict ACE Rules](#)

Best Practice ACE Rules

Best Practice ACEs are rules that internally perform checks against the structure of a Windows Installer package to enforce Microsoft best practices. Best Practice ACEs are similar to Microsoft ICEs. Both Microsoft ICEs (custom actions written by Microsoft used during validation) and Best Practice ACEs (custom actions included with ConflictSolver used during conflict analysis) are used to determine if an installation package is built according to Windows Installer standards.

When testing an installation package against a Best Practice ACE Rule, there is no target package involved. The source installation package is tested to see if it meets the standards defined in the ACEs.

Conflict ACE Rules

Conflict ACE Rules are used to detect conflicts between two Windows Installer packages. You can use Conflict ACE Rules to detect conflicts between one or more Windows Installer source packages and one or more target packages in the Application Catalog. Conflict evaluation is done for each source package against each target package.



Note • Multiple source package selection is only supported for internal packages (packages which have already been imported into the Application Catalog).

Check Conflicts Across All Source and Target Packages

In order to provide a more meaningful conflict analysis, you can choose to have ConflictSolver perform conflict analysis of each source package against every other source package and each target package against every other target package. This option is enabled by selecting the **Check conflicts across all source and target packages** option on the **Rules** tab of the ConflictSolver **Options** dialog box. By default, this option is not selected.

Connecting to Existing Application Catalogs

From ConflictSolver, you can choose to connect to a standalone Application Catalog or the AdminStudio Enterprise Server Application Catalog.

For more information, see [Connecting to an Existing Application Catalog](#).

Best Practices for Conflict Detection

When using ConflictSolver, the most database-intensive (and therefore most time consuming) tasks that you perform are importing packages and performing conflict detection. The time needed to complete these tasks depends upon how you have configured ConflictSolver and the steps that you follow. To ensure optimal performance of ConflictSolver during package import and conflict detection, we recommend that you incorporate the following best practices:

- **Plan Your Import**—If you are not using user-defined ACE rules, specify that you do not want to import binary data and custom tables. Also, specify that you do not want to check for validation errors or conflicts during import, and use the bulk import command line process. See [Plan Your Import](#).
- **Plan Your Conflict Detection**—Perform conflict detection on internal packages only, and limit the scope of conflict detection by choosing to run only the ACEs that are relevant to your organization. See [Plan Your Conflict Detection](#).
- **Perform Conflict Detection**—Follow this sequence of steps to quickly and efficiently perform conflict detection. See [Perform Conflict Detection](#).

Plan Your Import

Planning your import consists of the following steps:

- [Check and Configure Your Import Options](#)
- [Use the Command Line / Bulk Import Option of ConflictSolver](#)

Check and Configure Your Import Options

Before initiating an import, you should review the following options on the **Import** tab of the ConflictSolver **Options** dialog box:

- **Import Binary Data & Import Custom Tables**—If you do not expect to run user-defined ACE rules, then you do not need to import either binary data or custom tables.
- **Check for Validation Errors and Check for Conflicts**—Checking packages for validation at the time of import is often unnecessary. Also, conflict detection at the time of package importation will become an exponentially slow operation as the size of your Application Catalog increases, since this option causes the package to be conflict checked against all existing packages in the Application Catalog.
- **Check for Windows Integrity**—Select the **PackageExpert** option on the **Import** tab if you want to run Vista quality validation tests and Windows Installer validation tests on newly imported packages. See [Identifying and Resolving Package Errors Using PackageExpert](#).

Generally, unless it supports some specific business case, deselect the five import options on the **Import** tab of the ConflictSolver **Options** dialog box.

Use the Command Line / Bulk Import Option of ConflictSolver

ConflictSolver supports a bulk import process, which allows you to import multiple installer packages (such as MSI, MSM, MST, MSP, OSC and NCP files) at once without any user intervention. You can set up these files and have them run overnight or on an unattended workstation. By doing this, you minimize the need for someone to monitor the import process, freeing that person up to perform other tasks.

Also, if you use Bulk Import to import packages before they have been conflict checked, ConflictSolver performance is improved because it is quicker to perform conflict detection between internal installation packages than between internal packages and an external MSI package.

Performing a Bulk Import

To perform a bulk import, use the following command line:

```
iscmde.exe -I -C"c:\mypackages\myconfig.ini"
```

If you are using a configuration file, the only necessary parameters to pass at the command line are -I and -C"configuration_file_name.ini". You can include all other parameters inside the INI file. The following INI file is used to import packages into an SQL Server Application Catalog for a named user:

```
[General]
DatabaseType=SQL
LogFile=c:\temp\importlog.txt
MSIFile=3
MSMFile=1

[SQL]
Server=ConflictSolverSQL2K
UserID=Admin
Password=mypassword
Database=AdminStudio70

[MSIFile-1]
File=\\server\Data1.msi
Transform1=\\server\Data1a.mst
Transform2=\\server\Data1b.mst
Group=OfficeApps

[MSIFile-2]
File=\\server\Data2.msi
Group=OfficeApps\Secondary

[MSIFile-3]
File=\\server\Data3.msi

[MSMFile-1]
File=\\server\CrystalReports.msm
```

For more information on command line options, see [Application Manager Command-Line Functionality](#) and [Application Manager Configuration File](#).

Organizing Your Bulk Import

If a package has not been conflict checked, import it into a isolated (working) group, check it for conflicts, and only move it into another group when it is free of conflicts. For purposes of conflict detection, you would not use any of the other packages in this working group as one of the target packages because they are still in the process of being analyzed for conflicts against the rest of the Application Catalog (potentially by other users).

Plan Your Conflict Detection

Planning your conflict detection consists of the following steps:

- [Perform Conflict Detection on Internal Application Catalog Packages](#)
- [Check and Configure Your Conflict Options](#)

Perform Conflict Detection on Internal Application Catalog Packages

Checking for conflicts against external MSI packages is much slower than checking against a package already imported into the Application Catalog. Therefore, we recommend that you import all packages that you want to conflict check into the Application Catalog and select the **Internal Application Catalog package** option on the **Source Type** panel of the **Conflict Wizard**.

Check and Configure Your Conflict Options

ConflictSolver includes over twenty [Application Conflict Evaluators \(ACEs\)](#)—rules that allow ConflictSolver to identify conflicts between setups and between setups and operating system images.

Before beginning conflict detection, review the purpose of each of the ACEs. Then, on the **Rules** tab of the ConflictSolver **Options** dialog box, select only those conflicts that are appropriate for the packages you are checking. By limiting the scope of conflict detection, you will be improving the performance of the conflict detection process.

For a description of all ConflictSolver ACE rules, see [Application Conflict Evaluators \(ACEs\)](#).

Perform Conflict Detection

To ensure optimal performance of ConflictSolver during conflict detection, perform these steps in the following sequence:

- [Pre-Conflict Detection Planning](#)
- [Conflict Detection Processing](#)
- [Post-Conflict Detection Evaluation](#)

Pre-Conflict Detection Planning



Task: *To perform pre-conflict detection planning*

1. Create a new SQL Server Application Catalog database using the **Create** command on the AdminStudio Interface **Catalog** menu. See [Creating New Application Catalogs](#).



Note • If you have AdminStudio Enterprise Edition, you can also connect to the AdminStudio Enterprise Server Application Catalog. See [Connecting AdminStudio Client Tools to the AdminStudio Enterprise Server Application Catalog](#).

2. Use the **OS Snapshot Wizard** to generate snapshots for the desktop(s) used by your company. See [Capturing an OS Snapshot](#).
3. Use ConflictSolver to create the necessary groups in the database to match how you will distribute these packages. See [Adding Groups](#).
4. Import all OS Snapshot(s) into ConflictSolver using the bulk import process, and copy these snapshots into their appropriate groups. See [Application Manager Command-Line Functionality](#).
5. If you intend to do conflict detection with **ACE12** (meaning that you have access/rights to manipulate the MSI package in Tuner or InstallShield Editor to correct any conflicts that are found), then pre-import all Merge Modules that are used by your packages, ideally using the bulk import process. If you are not performing conflict detection on merge modules, you do not need to import merge modules.
6. Find out how these packages will be distributed and used in your enterprise, and then decide which ACEs you need to use during conflict detection.

Conflict Detection Processing



Task: *To perform conflict detection processing*

1. Perform validation. Validation errors may not be serious, but you should be aware of them. Because the results of the ConflictSolver **Validation Wizard** are not persisted in the database, you can use another product, like Microsoft Orca, to do your validation, if desired. See [Performing Validation](#).
2. Use the bulk import process to import your packages into a temporary working group.
3. Select your most critical, important application (perhaps for each group), and run validation and the source-only ACEs ([ACE04](#), [ACE05](#), [ACE06](#)) on it. If the package passes evaluation, then move it from the working group into its appropriate group. This package will represent the baseline against which all subsequent packages will be checked against.
4. Select your next package and run conflict detection with the ACE rules that you have identified as important against the group that this package will be assigned to. Evaluate the results and move the package into its appropriate group. Repeat the process. See [Identifying and Resolving Conflicts](#).

5. You should always be evaluating one package against a group of already approved packages for conflicts. Any resolutions that you perform will be against the package you are evaluating.

Post-Conflict Detection Evaluation

Once you are done performing conflict detection with all of your packages, you should have groups of packages which can co-exist with each other.

Scanning for Dependencies

Scanning for dependencies generates a list of all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog.

If dependency information has not already been generated by the Patch Impact Analysis Wizard, you can generate dependency information for a Windows Installer package by selecting it and selecting **Scan for dependencies** on the context menu.



Task: *To scan a package for dependencies:*

1. If the package that you want to scan is stored in the Software Repository, open Application Manager, select the package node, and select **Check out** from the context menu. The package is checked out.
2. Open ConflictSolver, select the package node, and select **Scan for dependencies** from the context menu.
3. Select the dependencies node to open the [Dependencies View](#), which lists all of the selected package's files that have dependencies with files used by other products or operating systems in the Application Catalog. The following information is displayed:

Option	Description
Files With Dependencies	Make selections from this list to further filter the output listing, or select (All) to display all dependencies.
File	Name of a file included in this product that is dependent upon a file used by another product or operating system in the Application Catalog.
Dependent File	Name of the file that this product's file is dependent upon.
Size	Size of the dependent file. This field is populated for file dependencies identified from the MSI package itself. For dependencies that were derived from the static scanning process, this field will be undefined.
Version	Version of the dependent file. For dependencies that were derived from the static scanning process, this field will be undefined.
Language	Language of the dependent file. For dependencies that were derived from the static scanning process, this field will be undefined.



Note • Regarding scanning for dependencies, note the following:

- If the **Only Display View Nodes With Data** option on the General tab of the ConflictSolver and Application Manager **Options** dialog box is selected, if you scan for dependencies and no dependencies are found, the **Dependencies** node will still not be displayed.
- Dependency information is also generated by the Patch Impact Analysis Wizard.
- If you want to perform validation, scan for dependencies, or conflict resolution on a package that is stored in the Software Repository, you must check out the package before you begin (because these actions may result in data in the package being changed).

Validating Packages

An important first step in conflict identification and resolution is to validate the original Windows Installer package. Validation compares the package against Microsoft ICEs (Internal Consistency Evaluators), custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards. ICE rules return the following types of messages:

- **Errors**—Database authoring that causes incorrect behavior. An example of an error would be duplicate component GUIDs, which would cause the Installer to incorrectly register components.
- **Warnings**—Database authoring that can cause incorrect behavior. Warnings can also report unexpected side-effects of database authoring. An example of a warning would be entering the same property name in two conditions that differ only by the case of letters in the name. Because the Installer is case-sensitive, the Installer treats these as different properties.
- **Informational**—Information from the ICE that do not indicate a problem with the database. Often they are information about the ICE itself, such as a brief description. They can also provide progress information as the ICE runs.



Note • For a full listing and description of all Microsoft ICEs, see [ICE Reference](#) in the Microsoft MSDN Library.

Ideally, all packages that you import into ConflictSolver will pass validation, but realistically that will not be the case. ConflictSolver allows you to import any Windows Installer package into ConflictSolver, regardless of whether it has passed validation or not. However, should you import a Windows Installer package that does not pass validation, potential issues may arise due to the package not having been authored to Microsoft standards.

You can choose to validate an MSI package either before import, during import, or after import. If you choose to validate the package after it has been imported into the Application Catalog, those validation results are persisted, are listed on the **Validation** view, and are included in the Package Report (see [Generating Package Reports](#)).

Tasks associated with validation include:

- [Performing Validation](#)
 - [Validating Before Import](#)
 - [Validating During Import](#)
 - [Validating After Import](#)
- [Changing the Default Validation File](#)
- [Excluding Specific ICEs from Execution During Validation](#)
- [Handling Invalid Windows Installer Packages](#)
- [About ICE43, ICE50, and ICE57 Validation Rules for Shortcuts](#)

Performing Validation

You can choose to validate an MSI package before it is imported into the Application Catalog, during import, or after it has been imported:

- **Before Import**—You can use the Validation Wizard to validate an MSI package before it has been imported into the Application Catalog.
- **During Import**—If the **Check for Validation Errors** option is selected on the Import Tab of the **Options** dialog box, validation will automatically be performed when you use the [Import Wizard](#) to import an MSI package. Behind the scenes, the Validation Wizard checks the package for validation errors, and if the package fails validation, it will not be imported into the Application Catalog.
- **After Import**—After an MSI package has been imported into the Application Catalog, you can use the Validate Package function to perform validation.

Whether you perform validation before, during, or after import, validation results are displayed in the **Validate** tab of the Output Window. However, if you perform validation after an MSI package has been imported into the Application Catalog, those validation results are persisted, and can be viewed in the Validation View and are included in the Package Report.


Validating Before Import



Before importing a package into the Application Catalog, you may want to perform validation to ensure that the package is built according to Windows Installer standards.



Task: *To validate a Windows Installer package prior to import:*

1. Select **Validation Wizard** from the **Conflicts** menu. The **MSI Source Information Panel** of the **Validation Wizard** opens.
2. Specify or browse to the Windows Installer package you want to validate.

3. If necessary, click the New button () and add any associated transforms to include in the validation. For multiple transforms, you can move the transforms up and down to change the order in which they are applied to the base package (which can affect validation).
4. Click **Next**. The **Summary Panel** opens.
5. Review your options in the Summary Panel. Click **Finish** to begin validation.

As validation occurs, messages are listed in the **Output** tab of the **Output Window**. When validation is complete, output appears in the **Validate** tab of the **Output Window** in table format. Each table row lists an icon to indicate whether it is an Error () or a Warning (), the name of the ICE that generated it, and a brief description of what caused it to occur.

6. To view or print out a list of these validation errors for review, click in the **Output** tab of the **Output Window**, select **Select All** from the context menu, and then copy and paste the text into another application.




Tip • Should you find any validation errors, you may be able to correct them using Tuner or InstallShield Editor.



Note • If you use the *Validate Package* function to validate a package after you have imported it into the Application Catalog (instead of performing validation before it is imported), you can view validation results in the Validation View and include validation results in the Package Report. See [Validating an Imported Package](#) and [Generating Package Reports](#).

Validating During Import

While you are importing a package into the Application Catalog using the [Import Wizard](#), you can choose to also perform validation to ensure that the package is built according to Windows Installer standards. To do this, you would select the **Check for Validation Errors** option on the **Import** tab of the **Options** dialog box.

From this point, when you import a package using the Import Wizard, validation will automatically be performed. Behind the scenes, the Validation Wizard checks the package for validation errors and reports those results in the **Validation** tab of the Output Window. If the package fails validation (by generating an Error ) , the package will not be imported into the Application Catalog.



Tip • Should you find any validation errors, you may be able to correct them using Tuner or InstallShield Editor.



Task: *To validate a package during import:*

1. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Open the **Import** tab.
3. Select the **Check for Validation Errors** check box.

4. Click **OK**.



Note • If you validate a package during import, and want to view or print out a list of validation errors for review, click in the **Output** tab of the **Output Window**, select **Select All** from the context menu, and then copy and paste the text into another application. However, if you use the **Validate Package** function to validate a package after you have imported it into the **Application Catalog** (instead of performing validation during import), you can view validation results in the **Validation View** and include validation results in the **Package Report**. See [Validating an Imported Package](#) and [Generating Package Reports](#).

Validating After Import

Before performing conflict detection on an imported Windows Installer package, it is a good idea to perform validation by using the **Validate Package** function available on the **Product View** to ensure that the package is built according to Windows Installer standards.

The **Validate Package** function includes the following tasks and options:

- [Validating an Imported Package](#)
- [Viewing ICE Error Information](#)
- [Suppressing an ICE Error](#)
- [Deleting an ICE Error](#)

Validating an Imported Package

You can validate an imported package by using the **Validate Package** function on the **Product View**.

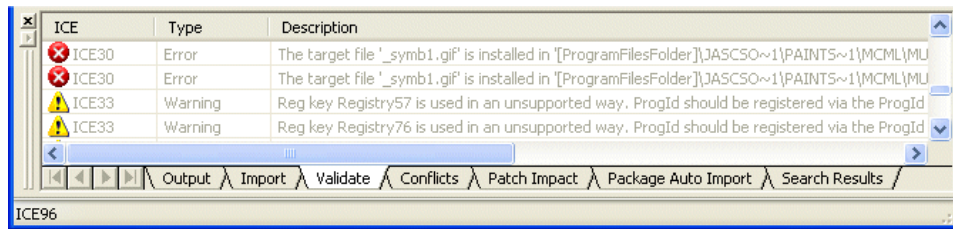


Task: *To validate an imported Windows Installer package:*

1. Launch **ConflictSolver**.
2. On the **Product View**, select the Windows Installer package that you want to validate and select **Validate Package** from the context menu.

Validation can only be performed on Windows Installer .msi packages. If you select another type of package in the **Application Catalog**, **Validate Package** will not be available on the context menu for that package.

As validation occurs, messages are listed in the **Output** tab of the **Output** window. When validation is complete, output appears in the **Validate** tab of the **Output** window in table format. Each table row lists an icon to indicate whether it is an Error (❌) or a Warning (⚠️), the name of the ICE that generated it, and a brief description of what caused it to occur.

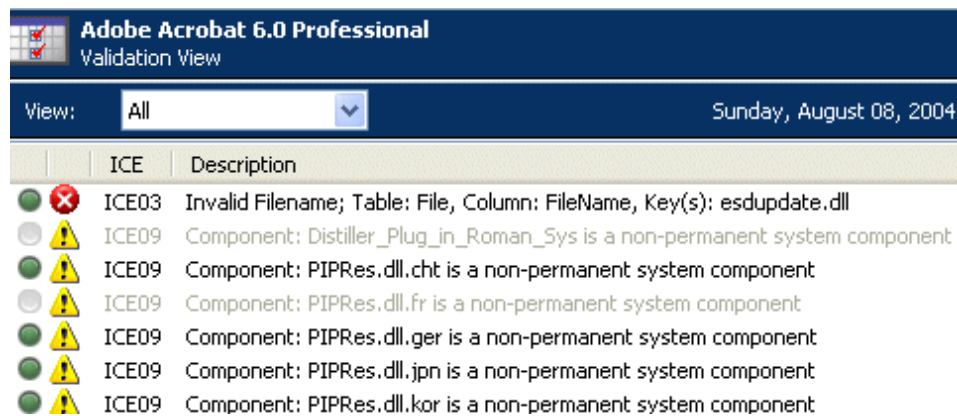


3. To view or print out a list of these validation errors for review, you have several options:

- Click in the **Output** tab of the **Output Window**, select **Select All** from the context menu, and then copy and paste the text into another application.
- Select the package and then select **Package** on the **Reports** menu or select **Package** under **Reports** on the context menu. See [Generating Package Reports](#) for more information.

Viewing ICE Error Information

After an imported package has been validated, detailed information on all of the ICE Errors and Warnings that were found can be viewed on the **Validation View**. Each ICE error that was found during validation is listed, with an icon to indicate whether it is an Error (✖) or a Warning (⚠), the name of the ICE that generated it, and a brief Description of what caused it to occur.



You can also view the full text of the description, add explanatory text, and suppress or enable each ICE error on the Properties dialog box.



Task: To view ICE error information:

- Perform the steps listed in [Validating an Imported Package](#).
- On the **Product View**, expand the tree for the validated Windows Installer package and select the **Validation** node. The **Validation View** opens.

3. To easily view the full text of the ICE error description, select an ICE Error or Warning number and then select **Properties** from the context menu. The **Properties** dialog box opens listing the ICE Error Number and its full **Description**.
4. To add a comment to this ICE error, add text in the **Explanation** text box.
5. To set the **State** of an ICE, select **Suppressed** or **Enabled**. See [Suppressing an ICE Error](#) for more information.
6. Click **OK** to close the **Properties** dialog box.



Note • The Validation View displays the ICE error information that was generated by using the Validate Package function available on the Product View for packages that have already been imported into the Application Catalog. Validation results generated for a package before it was imported into the Application Catalog or during import are not persisted and are not displayed on the Validation View. See [Validating After Import](#)

Suppressing an ICE Error

If you do not want ConflictSolver to check for a particular ICE error during subsequent validations of a package, you can choose to suppress it. When you suppress an ICE error, it is listed in gray on the Validation View and is not executed in subsequent validations. You might choose to suppress an ICE error that is a known issue at your organization which does not need additional corrections.

Suppressed ICE errors are still listed in the Package Report for that package. To remove an ICE error from the Package Report, you need to Delete it rather than Suppress it. See [Deleting an ICE Error](#) and [Generating Package Reports](#).



Task: [To suppress an ICE error:](#)




1. Perform the steps listed in [Validating an Imported Package](#).
2. On the **Product View**, expand the tree for the validated Windows Installer package and select the **Validation** node. The **Validation View** opens.
3. Select the ICE error that you want to suppress and then select **Properties** from the context menu. The **Properties** dialog box opens.
4. Under **State**, select **Suppressed**.



Note • You can also choose to enable or suppress an ICE error by selecting the ICE error on the **Validation View** and then selecting **Enable** or **Suppress** from the context menu. When using the context menu to set an ICE error's state, you can perform the operation on multiple ICE errors at once. Use the Shift key to select multiple contiguous items in the list, and use the Ctrl key to select multiple non-contiguous items.





5. You could choose to document the reason that you are suppressing this ICE error in the **Explanation** text box.
6. Click **OK** to close the **Properties** dialog box.

On the **Validation View**, the ICE error that you just suppressed is now listed in gray:

	ICE	Description
	ICE03	Invalid Filename; Table: File, Column: FileName, Key(s): esdupdate.dll
	ICE09	Component: Distiller_Win_NT_System_files is a non-permanent system component
	ICE09	Component: Distiller_Win9X_System_files is a non-permanent system component

Suppressing a ICE Error vs. an ICE Rule

When you suppress an ICE error, you are just suppressing that particular error; you are not suppressing subsequent checks for an entire ICE rule during subsequent validations of that package. For example, if you suppress one ICE57 error, all of the other ICE57 errors are not suppressed:

	ICE57	Component 'Elements_Bin_UserRegistry' has both per-user and per-machine data with a per-machine KeyPath.
	ICE57	Component 'Elements_Bin_EXE' has both per-user and per-machine data with a per-machine KeyPath.
	ICE57	Component 'Elements_Desktop_Shortcut' has both per-user and per-machine data with a per-machine KeyPath.
	ICE57	Component 'PDFMakerX' has both per-user and per-machine data with a per-machine KeyPath.

To suppress all ICE errors for an ICE rule, do one of the following:

- **To suppress an ICE rule for one package**—Perform the steps listed in [Validating an Imported Package](#), and then go to the [Validation View](#) and suppress all ICE errors found for that ICE rule.
- **To suppress an ICE rule for all packages in the Application Catalog**—Enter the ICE rule number on the **Validate** tab of the **Options** dialog box. See [Excluding Specific ICEs from Execution During Validation](#).

Deleting an ICE Error

You can permanently delete an ICE error from subsequent execution and reporting for that package. ICE errors that are deleted are no longer listed in the [Validation View](#) and will not be listed in the Package Report.



Task: *To delete an ICE error:*

1. Perform the steps listed in [Validating an Imported Package](#).
2. On the **Product View**, expand the tree for the validated Windows Installer package and select the **Validation** node. The **Validation View** opens, listing all ICE errors for the selected package.
3. Select the ICE number of the specific ICE error(s) you want to delete and select **Delete** from the context menu. You can delete multiple ICE errors at once. Use the **Shift** key to select multiple contiguous items in the list, and use the **Ctrl** key to select multiple non-contiguous items.
 - If you are deleting an ICE error that is currently Suppressed, you will be asked to confirm the deletion.
 - If you are deleting an **Enabled** ICE error, no confirmation message will appear.


The deleted ICE error will no longer be listed in the **Validation View** or on the Package Report for that package. You have permanently suppressed the selected ICE error, and ConflictSolver will not execute that ICE error during subsequent validations of this package.

Changing the Default Validation File

Validation involves comparing a Windows Installer package against a known criteria to identify deviations from those rules. By default, ConflictSolver compares packages against the Full MSI Validation Suite. This suite contains a comprehensive set of [Internal Consistency Evaluators \(ICEs\)](#)—guidelines created by Microsoft to ensure an installation package works correctly with the Windows Installer engine. In the overwhelming majority of cases, this is the file you will want ConflictSolver to use. However, there may be times you want to compare your base package against a different validation file, depending on your needs.



Task: *To specify a different validation file to use:*

1. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Open the **Validate** tab.
3. In the **MSI Input files** and **MSM Input files** areas, click the Browse button () next to the **Cub File** field to locate the validation file (.cub) that you would like to use.
4. To permanently exclude the execution of specific ICEs, list them in the **ICE Rules** field, separated by semicolons (;). Otherwise, all rules will be used. See [Excluding Specific ICEs from Execution During Validation](#) for more information.

Excluding Specific ICEs from Execution During Validation

You can permanently exclude the execution of specific ICEs that are listed in the selected .cub file.



Task: *To exclude the execution of specific ICEs during validation:*

1. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Open the **Validate** tab.
3. In the **MSI Input files** and **MSM Input files** areas, list the ICEs that you want to exclude in the **ICE Rules** field, separated by semicolons (;), such as:

ICE07; ICE09; ICE22

All of the ICEs listed in the specified .cub files will be executed during validation except for the ICEs entered here.

Handling Invalid Windows Installer Packages

Ideally, all Windows Installer packages will pass validation. Realistically, many will fail. When a package fails validation, it means the package was not built to Microsoft's specifications. It does not mean the installation does not work. However, there are a few things you can do when your package has validation errors:

Table 13-3 • Validation Error Solutions

Solution	Explanation
Use Tuner to correct validation errors.	This involves opening the base package using Tuner and creating a transform file which contains your corrections. The same validation tool is available in Tuner.
Contact the installation vendor.	The company that created the installation (usually the same company that created the software) may be able to resolve the validation issues and provide you with a valid setup. Be sure to provide the validation report to vendors so they know where to focus.
Reconsider using the application.	Although it might be an extreme reaction to an invalid package, there may be compelling reasons not to use an installation not built to Microsoft guidelines.
Ignore the problems and import the package into ConflictSolver anyway.	This is probably the most likely scenario. The invalid installation may not be worth trying to fix, or even have errors that you are concerned about. ConflictSolver allows you to import an installation package regardless of its validity; just know that there could be problems in the future in doing so. From a practical standpoint, this may be your best option.

About ICE43, ICE50, and ICE57 Validation Rules for Shortcuts

Each entry in the **CreateLink** section of the .inc file is converted into an entry in the Shortcut table. The exact properties of the shortcut depends upon the information in the **CreateLink** line as well as the nature of the target file itself.

ICE 43, ICE 50, and ICE 57 are the most common validation rules for shortcuts.

Shortcut Types

The primary distinction between shortcuts is advertised vs. non-advertised. Here are two reasons why it is preferable to create advertised shortcuts:

- Advertised shortcuts are triggers for MSI's self-repair mechanism.
- Non-advertised shortcuts are intended for a per-user context only:
 - The target file must be a file installed in a user-specific directory.
 - The keypath of the component containing the target file must be a user-specific registry value.

Conversion from CreateLink entries to Shortcut Table Entries

The .inc converter will always try to create advertised shortcuts for every CreateLink line found in the .inc file. However, not every CreateLink line can be converted into an advertised shortcut.

In order to create an advertised shortcut, the information in the CreateLink line must meet all of the following requirements:

- The target file must be the keypath of its component. This means that the target file must be listed in the .inc file list. The converter will create a new component for a non-PE (portable executable) target files, so that it is guaranteed to become the keypath of the component. (Normally, for each target directory, non-PE files are grouped together into one component).
- The target file must contain an icon.

In general, this means that as long as the target file contains an icon, the converter will be able to create an advertised shortcut for it. However, whenever a shortcut cannot be advertised, the converter does the following:

- It creates a “catch all” component (if not yet created) named ShortcutsComponent. It also creates an HKCU registry entry in the Registry table, and that entry is used as the keypath for the ShortcutsComponent. This is done to avoid ICE43.
- A new shortcut entry is created in the Shortcut table, associated with the ShortcutsComponent.

Identifying and Resolving Conflicts

Conflict identification occurs if you import a package into ConflictSolver using the [Import Wizard](#), providing the **Check for Conflicts** option has been selected in the **Import** tab of the **Options** dialog box. ConflictSolver checks for conflicts using the default rules specified in the **Rules** tab of the **Options** dialog box.

You can also identify conflicts using the [Conflict Wizard](#). From the Source Type Panel, you can select whether you want to check an external Windows Installer package against packages in the Application Catalog, or check one or more packages already in the Application Catalog against others in the Application Catalog.

The following topics explain how to perform conflict identification and resolution:

Table 13-4 • Conflict Identification and Resolution Tasks

Section	Topics
Best Practices for Conflict Detection	<ul style="list-style-type: none">• Plan Your Import• Plan Your Conflict Detection• Perform Conflict Detection
Setting Options	<ul style="list-style-type: none">• Changing Default Conflict Types Checked• Changing Resolution Options

Table 13-4 • Conflict Identification and Resolution Tasks (cont.)

Section	Topics
Checking for Conflicts	<ul style="list-style-type: none">• Checking for Conflicts Using the Conflict Wizard• Checking for Conflicts During Import• Viewing Identified Conflicts in the Conflicts View• Running Conflict Identification Again• Deleting Persisted Conflict Data• Conflict Persistence
Resolving Conflicts	<ul style="list-style-type: none">• Conflict Resolution Process• Automatically Resolving Conflicts• Manually Resolving Conflicts• Resolving Conflicts Directly in Windows Installer Packages• Using Transforms for Conflict Resolution• Reimporting Packages after Successful Conflict Resolution

Using Multiple Source Packages in an Analysis

When performing conflict analysis between packages in the Application Catalog, you may choose to select either one or multiple source packages and multiple target packages. Conflict analysis is then performed for each source package against each target package.

Also, in order to provide a more efficient conflict analysis, you can choose to have ConflictSolver perform conflict analysis of each source package against every other source package and each target package against every other target package. This option is enabled by selecting the **Check conflicts across all source and target packages** option on the **Rules** tab of the **Options** dialog box. By default, this option is not selected.

Setting Options

You can set the following ConflictSolver conflict identification and resolution options:

- **Changing Default Conflict Types Checked**—Select the ACE rules that will be selected by default during conflict detection.
- **Changing Resolution Options**—Specify whether you want to save resolutions in the original MSI package or in a transform file.

Changing Default Conflict Types Checked

In the **Enabled Predefined and Custom Rules** area of the **Rules** tab of the **Options** dialog box, you can select the default set of ACEs that will be included in conflict analysis.

ACEs associated with unselected boxes will not be performed by default during conflict identification. However, you can override these settings in the **Summary** panel of the **Conflict Wizard**.



Task: *To change default conflict types checked:*

1. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Open the **Rules** tab.
3. In the **Enabled Predefined and Custom Rules** area, select the **Show best practice rules** option. The **Best Practices** ACE rules tree is displayed.
4. Expand the tree and select the Best Practice ACE rule conflicts you want to check for from the list.
5. Select the **Show conflict detection rules** option. The **Conflict Types** ACE rules tree is displayed.
6. Expand the tree and select the **Conflict Types** ACE rule conflicts you want to check for from the list.
7. When performing conflict analysis using multiple source packages and one or more target packages, ConflictSolver will evaluate each source package against each target package. However, if you want ConflictSolver to also perform conflict analysis of each source package against every other source package and each target package against every other target package, select the **Check conflicts across all source and target packages** option.
8. Click **OK**.

The ACE rules associated with each conflict type are listed in the following table:

Table 13-5 • Best Practice ACE Rules

ACE Type	Conflict Type	Associated Rules
Best Practice ACEs	Components	ACE04, ACE05, ACE06,
	Merge Module Integrity	ACE26, ACE36
	Recommended Rules	ACE25, ACE27, ACE28, ACE29, ACE31, ACE32, ACE33, ACE34, ACE35
Conflict ACEs	Component	ACE02, ACE09, ACE30
	File Extensions	ACE17
	Files	ACE03, ACE07, ACE08, ACE12, ACE23
	INI Files	ACE14, ACE21, ACE22
	NT Services	ACE16

Table 13-5 • Best Practice ACE Rules

ACE Type	Conflict Type	Associated Rules
	ODBC Resources	ACE15
	Product Properties	ACE18, ACE19, ACE20
	Registry	ACE10, ACE24
	Shortcuts	ACE13

Changing Resolution Options

You can change resolution options on a product-by-product basis.



Task: *To change resolution options:*

1. From the **Product View**, select the product for which you want to change resolution options.
2. Click on the plus sign to expand the tree listing and select the **Conflicts** node. The **Conflicts** view opens.
3. Ensure the **Summary** view is selected from the **Conflict Information** list.
4. In the **Resolution Options** area, click **Edit**. The **Resolution Options** dialog box opens.
5. Select whether you want to save resolutions in the **original MSI package** or in a **transform file**. If you select a transform file, browse to or specify it.
6. Click **OK**.

Any resolutions you make are stored in the specified transform (if selected) or in the original package. By selecting the **Perform resolutions against a transform file** option in the **Resolution** tab of the **Options** dialog box, you can set the default behavior to store resolutions in a transform.



Note • You can only modify resolution options if persisted conflict data exists for a product.

Checking for Conflicts Across All Source and Target Packages

When performing conflict analysis using multiple source packages and one or more target packages, ConflictSolver will evaluate each source package against each target package.

However, if you want ConflictSolver to also perform conflict analysis of each source package against every other source package and each target package against every other target package, select the **Check conflicts across all source and target packages** option on the [Rules Tab](#) of the **Options** dialog box.



Note • If performing conflict analysis with multiple source packages, and if this option is selected, both [ACE21](#) and [ACE22](#) are automatically evaluated when either one is selected. If only one of them is evaluated, then there is the possibility that conflicts will not be detected.

Checking for Conflicts

The following topics are included in this section:

- [Checking for Conflicts Using the Conflict Wizard](#)
- [Checking for Conflicts During Import](#)
- [Viewing Identified Conflicts in the Conflicts View](#)
- [Running Conflict Identification Again](#)
- [Deleting Persisted Conflict Data](#)
- [Conflict Persistence](#)

Checking for Conflicts Using the Conflict Wizard

The Conflict Wizard can be used to identify conflicts between an external Windows Installer package and ones already in the Application Catalog, or between one or more packages in the Application Catalog.

Identifying Conflicts with an External Package





The ConflictSolver Wizard allows you to run conflict checks against external MSI packages and packages imported into the ConflictSolver Application Catalog Database. ConflictSolver supports external package conflict checking for User Defined Custom/Source Only Packages ACEs. The Source package can be selected from the ConflictSolver Application Catalog Database or from an external MSI package. See [Creating a Custom/Source Only Packages ACE](#) for more information.

However, ConflictSolver does not support external package conflict checking for Custom/Source and Target Packages ACEs. Both the Source and Target Packages must be selected from the ConflictSolver Application Catalog Database.



Task: *To identify conflicts between an external Windows Installer package and packages already in the Application Catalog:*

1. Select **Conflict Wizard** from the **Conflicts** menu. The **Welcome Panel** opens.
2. Click **Next**. The **Source Type Panel** opens.
3. Select **Windows Installer** package. Click **Next**. The **MSI Source Information Panel** opens.
4. Click **Browse** to locate the MSI file you want to check against packages in the Application Catalog.

5. Should that package require transforms, click the New button () above the **Transforms** area. Use the Browse button (...) to locate the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up () and Move Down () buttons. If you need to delete a transform you have added, select the transform and click the Delete button ().
6. Click **Next**. The **Target Information Panel** opens.
7. Select the packages in the Application Catalog you want to compare the selected package against. You can also select entire groups, which will compare the package against all packages and OS Snapshots in that group.
8. Click **Next**. The **Conflict Rules Panel** opens.
9. Select the types of conflicts you want to identify and click **Next**. The **Summary Panel** opens.
10. Review the options selected in the Summary Panel and click **Finish** to begin the conflict identification process. Progress appears in the **Output** tab of the **Output Window**.

When the conflict analysis is complete, all of the Errors and Warnings that were generated are listed in a table format in the **Conflicts** tab of the **Output Window**.

If a row is active, you can double-click on it to view that row's associated table. The **Tables View** is launched and the table and/or table cells that are causing the problem are highlighted in red. If a row is grayed out, it indicates that the table cannot be viewed in the ConflictSolver (perhaps because it is in an external package). For information on resolving conflicts, see [Resolving Conflicts](#).

Identifying Conflicts Between Existing Packages



Task: *To identify conflicts between packages already in the Application Catalog:*

1. Select **Conflict Wizard** from the **Conflicts** menu. The **Welcome Panel** opens.
1. Click **Next**. The **Source Type Panel** opens.
2. Select **Internal Application Catalog package** and click **Next**. The **Source Package Panel** opens.
3. Select the package(s) you want to compare against other package(s) in the Application Catalog.

When performing conflict analysis using multiple source packages and one or more target packages, ConflictSolver will evaluate each source package against each target package. However, if you want ConflictSolver to also perform conflict analysis of each source package against every other source package and each target package against every other target package, select the **Check conflicts across all source and target packages** option on the **Rules** tab of the **Options** dialog box.

4. Click **Next**. The **Choose Action Panel** opens.
5. To include Best Practice ACE rules in this conflict analysis, select the **Evaluate source package against best practice rules** option.
6. To include Conflict ACE rules in this conflict analysis, select the **Detect conflicts between source package and other packages** option.

7. To display the **Conflict Resolution** dialog box after conflict detection, select the **Run Resolution** option.

From the **Conflict Resolution** dialog box, you can access information pertaining to the last execution of the Conflict Wizard. This persisted conflict data allows you to view when the last execution was performed, the products that it was run against, the ACE rules used, and conflicts discovered. The dialog box also provides a list of updated, deleted, or added products that may necessitate performing conflict identification again.
8. To generate a conflict detection analysis report containing a summary of the conflicts that were detected, and a detailed description of each identified conflict including how this conflict will be resolved, select the **Generate Reports** option.
9. Click **Next**. If you selected the **Evaluate source package against best practice rules** option on the **Choose Action Panel**, the **The Best Practice Rules Panel** opens.
10. Select the Best Practice Rules that you want to evaluate in conflict analysis. By default, the Best Practice ACEs that are selected on the **Rules** tab of the **Options** dialog box will be selected.
11. Click **Next**. If you selected the **Detect conflicts between source package and other packages** option on the **Choose Action Panel**, the **Conflict Rules Panel** opens.
12. Select the Conflict Rules that you want to evaluate in conflict analysis. By default, the Conflict Rules ACEs that are selected on the **Rules** tab of the **Options** dialog box will be selected.
13. Click **Next**. If you selected any Conflict Rules to evaluate, the **Target Information Panel** opens.
14. Select all the packages you want to compare the source package(s) against. (The Target Information Panel excludes all packages that you selected on the Source Package Panel.) Click **Next**. The **Summary Panel** opens.
15. Review the options selected in the **Summary Panel** and click **Finish** to begin the conflict identification process. Progress appears in the **Output** tab of the **Output Window**.

When the conflict analysis is complete, all of the Errors and Warnings that were generated are listed in a table format in the **Conflicts** tab of the **Output Window**.

If a row is active, you can double-click on it to view that row's associated table. The **Tables View** is launched and the table and/or table cells that are causing the problem are highlighted in red. If a row is grayed out, it indicates that the table cannot be viewed in ConflictSolver (perhaps because it is in an external package). For information on resolving conflicts, see [Resolving Conflicts](#).
16. If you selected the **Run Resolution** option on the **Choose Action panel**, the **Conflict Resolution** dialog box opens, listing the total number of conflicts that were identified and the number of products and rules that were checked. See [Resolving Conflicts](#) for more information.

Checking for Conflicts During Import



Task: *To check for conflicts during import:*

1. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
2. Open the **Import Tab**.

3. Select the **Check for Conflicts** check box.
4. Open the **Conflicts** tab.
5. Select the conflict types you want checked by default when importing packages.
6. Click **OK**.


From this point, when you import a package into the Application Catalog, the package will automatically be checked for conflicts against all packages already in ConflictSolver using the default ACE rules selected in the **Conflicts** tab of the **Options** dialog box.

Depending on how many packages already in ConflictSolver, and depending on the number of ACE rules selected by default, this can take a significant amount of time.

Viewing Identified Conflicts in the Conflicts View



Task: To view identified conflicts:

1. From the **Product View**, select the product for which you want to view identified conflicts.
2. Select the Conflicts icon () to view the **Conflicts View** for the product.

The Summary view displays an overall summary of the conflicts identified, including the products this product was checked against for conflicts, the ACE rules used, and resolution options.

You can change the Conflict Information list at the top of the view to see individual conflict categories and perform resolution. See [Resolving Conflicts](#) for more information.

Consolidated Conflicts Listing

When evaluating conflicts between a source package and multiple destination packages, the same conflict may be detected between the source package and more than one destination package. In the **Conflicts View** listing, these conflicts are consolidated and displayed as single entry whenever possible.

To see a list of all of the target packages that detected the same conflict, click the **Target Packages** button to view a list of the package on a pop-up dialog box.



Note • You can also view conflicts generated from the comparison of an external package with those packages already in ConflictSolver by selecting **Resolve External Package Conflicts** from the **Conflicts** menu. The resulting **Conflict Resolution** dialog box displays conflict results from the last conflict execution for an external package.

Running Conflict Identification Again

At times, it may be necessary to run conflict identification again on a package that you have previously checked for conflicts. This could be because other packages you previously compared this package against have changed. It may be because you have added additional packages to ConflictSolver.

There are two ways to accomplish this.

- If you want to run conflict identification again using the same options as the last time you ran conflict identification on the package, you can do so by clicking the Run Again button from the package's [Conflicts View](#).
- In all other cases, use the [Conflict Wizard](#) to perform conflict identification.



Note • When you run conflict identification for a package on which you have previously performed conflict identification, the persisted conflict data is deleted and replaced with the new conflict data.

Deleting Persisted Conflict Data



Task: *To delete persisted conflict data from a package:*

1. Right-click on the package in the **Product View**, point to **Delete**, and select **Persisted Conflict Information**.
2. Confirm the deletion.

Conflict Persistence

ConflictSolver automatically persists conflict information for packages in ConflictSolver, and for the last execution of the Conflict Wizard between an external package and packages already in ConflictSolver. This enables you to run conflict identification between packages without having to perform the resolution immediately.

Your workflow may, for example, dictate that you import several packages into ConflictSolver first (placing them in some sort of staging group) and then perform conflict identification on them all at once. Resolution of identified conflicts may be much later in the workflow. ConflictSolver assists you by retaining the conflict information until you are ready to work with it.

Each package in ConflictSolver includes a [Conflicts View](#), where this persisted conflict data can be accessed. You can see when conflict identification was performed, between which packages, using which ACE rules, and the result. You can also see which packages have been updated, added to, or deleted from ConflictSolver since conflict identification was performed.

Resolving Conflicts

The following topics are included in this section:

- [Conflict Resolution Process](#)
- [Automatically Resolving Conflicts](#)
- [Manually Resolving Conflicts](#)
- [Resolving Conflicts Directly in Windows Installer Packages](#)
- [Using Transforms for Conflict Resolution](#)
- [Reimporting Packages after Successful Conflict Resolution](#)



Caution • If you want to perform validation, scan for dependencies, or conflict resolution on a package that is stored in the Software Repository, you must check out the package before you begin (because these actions may result in data in the package being changed).

Conflict Resolution Process

After conflict analysis is performed, either by using the Conflict Wizard or by checking for conflicts during import, all of the Errors and Warnings that were generated are listed in the Conflicts tab of the Output Window in a table format. You can resolve these conflicts by clicking Resolve or Resolve All on the Conflicts View. However, the Conflict Resolution Process differs depending upon options you select on the Resolution Tab of the Options dialog box.

Review and Approve Conflicts

The selection you make in the Preview and approve all resolutions option on the Resolution Tab of the Options dialog box determines if you will be prompted to review resolutions before they are executed.

- If the **Preview** and **Approve all resolutions** option is selected, the Resolution Details dialog box appears when you click **Resolve** or **Resolve All** on the Conflicts View.

This dialog box lists the proposed CARD resolutions that will be made to the package. Only the CARD resolutions that are selected when you click **Approve** will be executed.
- If the Preview and approve all resolutions option is not selected, the Resolution Details dialog box will not appear, and all selected conflict resolutions will be executed.
- If the Resolution Details dialog box opens but you no longer want it to, select the Don't show this dialog again check box. This is equivalent to clearing the Preview and approve all resolutions option on the Resolution Tab of the Options dialog box.

User Input Option

If a CARD can be resolved in multiple ways, and if the Maximize user flexibility in selecting resolutions option is selected on the Resolution Tab of the **Options** dialog box, the CARD Resolution Options dialog box appears, prompting you to make a selection to determine how conflict resolution will proceed. The following CARD can be resolved in multiple ways:

- **CARD06**—ACE06 accumulates a list of potential files to be used for the KeyPath. The CARD Resolution Options dialog box appears, prompting you to choose one file from the list of potential files. See [CARD06](#) for more information.

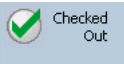
Conflict Resolution and the Software Repository

If a package is stored in the Software Repository, resolution and re-import behaviors (behaviors that modify the Application Catalog) are dependent upon a package's checked-in or checked-out status.

Resolution Behaviors

Because conflict resolution requires that the package be modified, if a package is stored in the Software Repository and is currently checked-out by another user, it cannot be modified, and therefore cannot be resolved. This resolution behavior is detailed in the following table.

Table 13-6 • Job Manager Resolution Behavior Regarding the Software Repository

Package State	Resolution Behavior
Package in Software Repository / Checked Out	Job Manager ignores the resolution process for these packages. The packages are already being edited by someone else.
Package in Software Repository / Not Checked Out	<p>The package will be checked out by the virtual user Job Manager Engine prior to the normal operation of the resolution process.</p> <div data-bbox="609 1308 732 1371"></div> <p>Job Manager Engine Friday, August 05, 2005 - 11:35 AM</p> <p>If the resolution process results in a no-operation or the user does not elect to re-import the package, then the check-out operation will be cancelled.</p>
Package Not in Software Repository	The resolution process will operate normally.

Re-Import Behaviors

When defining a Resolution Job Step, you have to indicate the reimport option for resolved packages. When defining a **Resolution** Job Step, you have the choice of three re-import options for resolved packages: re-import as a new version, re-import by overwriting the existing version, or do not re-import.

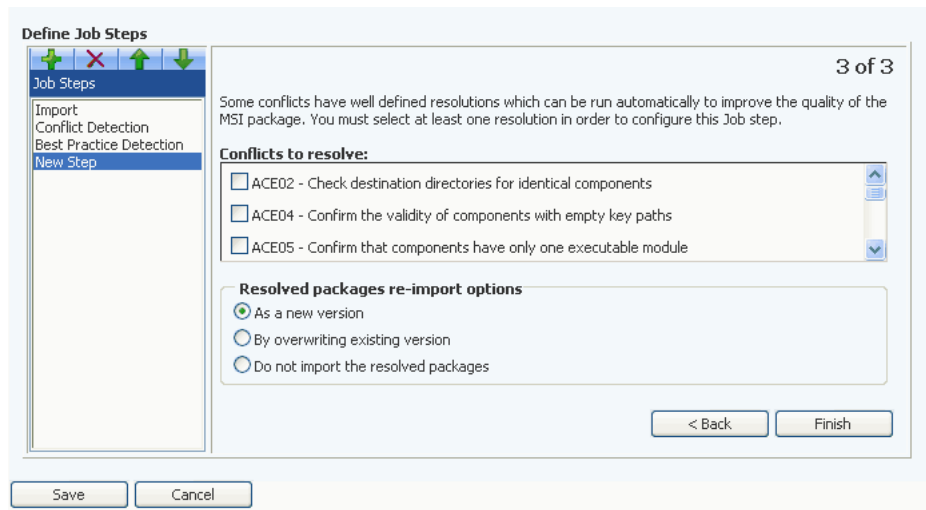


Figure 13-1: Reimport Options in Job Manager

The choice you make depends upon whether the package is stored in the Software Repository:

Table 13-7 • Resolved Package Re-Import Options

Resolved Packages Re-import Options	Package in Software Repository	Package Not in Software Repository
As a new version	Package will be re-imported and checked-in as a new version.	N/A Package cannot be imported as a new version unless it is in the Software Repository.
By overwriting existing version	Package will be re-imported and checked-in, replacing the existing version.	Package will be re-imported, replacing the original package.
Do not import the resolved packages	Package will not be re-imported.	Package will not be re-imported.

Automatically Resolving Conflicts

If the Conflict Wizard identifies conflicts, ConflictSolver may be able to resolve some of them automatically (ACE02, ACE04, ACE05, ACE06, ACE07, ACE15, ACE18, ACE19, and ACE20). You first need to view conflicts identified, and then select to resolve them.






Task: *To resolve an automatically resolvable conflict:*

1. From the **Product View**, select the product for which you want to view identified conflicts.
2. Select the **Conflicts View** for the product.
3. Select the category containing the automatically resolvable conflict from the **Conflict Information** list.
4. Expand the ACE containing conflicts that can be resolved.

The icon displayed next to the conflict indicates its status:

Table 13-8 •

Icon	Status
	Resolvable
	Resolved
	Unresolvable

5. Select the conflict you want to resolve and click **Resolve**. Alternatively, click **Resolve All** to resolve all resolvable conflicts for the selected conflict category.
6. For CARDS that can be resolved in multiple ways, the **CARD Resolution Options** dialog box appears, prompting you to make a selection to determine how conflict resolution will proceed. Make a selection and click **OK** to continue.



Note • CARDS can be resolved in multiple ways if the **Maximize user flexibility in selecting resolutions** option is selected on the **Resolution** tab of the **Options** dialog box.

7. If the **Preview and approve all resolutions** option on the **Resolution** tab of the **Options** dialog box is selected, the **Resolution Details** dialog box will appear, listing the proposed CARD resolutions that will be made to the package.

Select the CARD resolutions that you want to execute, and click **Approve** to execute them.

If the **Preview and approve all resolutions** option is not selected, the **Resolution Details** dialog box will not appear, and all selected conflict resolutions will be executed.

Resolutions are made to either the original Windows Installer package or in a transform (if the Perform resolutions against a transform file option is selected on the Resolution tab of the Options dialog box).

Manually Resolving Conflicts

Due to their complexity, some conflicts require manual resolution using Tuner or InstallShield Editor.

For information on manual solutions, consult the [Application Conflict Evaluators \(ACEs\)](#) topic or individual ACE topics. These topics can provide valuable information regarding what is causing the conflict, so you can use Tuner or InstallShield Editor to solve it.

Once you have resolved detected conflicts manually, you can reimport the package into ConflictSolver.

Resolving Conflicts Directly in Windows Installer Packages

ConflictSolver provides two methods for implementing conflict resolutions: directly in the Windows Installer package, or through the use of a transform.



Task: *To configure ConflictSolver to resolve conflicts directly in the original Windows Installer package:*

1. Select **Options** from the **Tools** menu to open the **Options** dialog box.
2. On the **Resolution Tab**, clear the Perform resolutions against a transform file check box.
3. Click OK to close the **Options** dialog box.

With this setting disabled, all conflict resolutions are by default performed directly on the base Windows Installer package. In either case, you can override this setting on a package-by-package basis by clicking Edit in the Summary section of the Conflicts View and editing the resolution options on the Resolution Options dialog box.

Using Transforms for Conflict Resolution

ConflictSolver provides two methods for implementing conflict resolutions: directly in the Windows Installer package, or through the use of a transform.



Task: *To configure ConflictSolver to use transforms for conflict resolutions:*

1. Click **Options** on the **Tools** menu. The **Options** dialog box opens.
2. Open the **Resolution Tab** and select the **Perform resolutions against a transform file** check box.
3. Click **OK** to close the **Options** dialog box.

With this setting enabled, all resolutions will by default be placed within a transform, rather than being made in the Windows Installer package. In either case, you can override this setting on a package-by-package basis by clicking Edit in the Summary section of the **Conflicts View** and editing the resolution options on the **Resolution Options** dialog box.

Reimporting Packages after Successful Conflict Resolution

Following resolution of conflicts, you should reimport the package into ConflictSolver. If you have performed automatic resolutions, you can do so by clicking the **Do you want to reimport this package?** link in the **Product View** for that package. The package (and the resolution transform if used) are reimported into ConflictSolver.

However, if you have performed manual resolutions to the package via a transform, you need to delete the package from ConflictSolver and then reimport the package with its externally created transform.

After reimporting a package, it is generally a good practice to run conflict identification again.

Generating Reports

Pre-Defined Reports

ConflictSolver includes several built-in reports to provide information on packages and conflicts identified, all accessible from the **Reports** menu. The included reports are:

Table 13-9 • Pre-Defined Reports List



Reports	Description
Package	<p>Provides a report including the following information:</p> <ul style="list-style-type: none">• Product Information—Lists all information that appears on the Product View for a package.• Conflict Information—Lists all conflicts found during the most recent Conflict Wizard execution, or the most recent Import (if conflict identification was performed).• Validation Information—Lists all validation errors and warnings found during the most recent validation execution for that package.  <p>Note • Validation information is persisted only for validations performed after the package was imported into the Application Catalog; validation results for validation performed during import are not persisted.</p> <ul style="list-style-type: none">• Extended Attributes—Lists user-defined information that appears on the Extended Attributes View for a package.
File	<p>The following information is displayed:</p> <ul style="list-style-type: none">• Product Information—Lists all information that appears on the Product View for a package.• Files—Lists all of the files that are included with the package. These files are also listed on the Files/Components View for that package.

Table 13-9 • Pre-Defined Reports List

Reports	Description
Registry	<p>The following information is displayed:</p> <ul style="list-style-type: none"> • Product Information—Lists all information that appears on the Product View for a package. • Registry—Lists any registry entries created or changed by the package. These registry entries are also listed on the Registry View for that package.
Crystal Reports	<p>The following reports in Crystal Report format are available:</p> <ul style="list-style-type: none"> • Crystal Reports Conflicts—Provides a report of all conflicts found during the most recent Conflict Wizard execution, or the most recent Import (if conflict identification was performed). This report is also available on a per-product basis by right-clicking on a product in the Product View. • Crystal Reports Files—Provides a comprehensive listing of all files in ConflictSolver, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View. • Crystal Reports Registry—Provides a comprehensive listing of all registry entries in ConflictSolver, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View.  <p>Note • The three Crystal Reports selections will be listed on the Report menu if the Display RPT Reports option on the General tab of the ConflictSolver Options dialog box is selected. To suppress the display of the Crystal Reports selections, clear this option.</p>

User-Defined Reports

ConflictSolver supports custom user-defined reports; however, the responsibility of creating them is up to individual users. Reports must be in [Crystal Reports](#) format (.rpt) and placed in the folder specified in the General tab of the **Options** dialog box, available from the Tools menu. These reports appear under the Reports menu under Crystal Reports. See [Creating Custom Reports](#) for detailed instructions.

Report Center Reports

In AdminStudio Enterprise Edition, you can view reports using the AdminStudio Enterprise Server Report Center. Available reports include:

- Catalog Summary Report
- Package Reports
- Package Contents Reports
- Consolidated Package Information
- Group Summary Report

For more information, see [Generating and Viewing Reports in Report Center](#).

Generating Package Reports

A Package Report lists general package information, extended attributes for that package, and any validation and conflict information that exists.



Task: *To generate a Package Report:*

1. In the **Products View**, select the root node of the product that you want to generate a Package Report for.
2. Click **Package** on the **Reports** menu.

or

Click **Package** under **Reports** on the context menu.

You are prompted to select a location where you want to save the report.
3. Select a location and click **Save**. The Package Report opens in a browser window displaying the following information:
 - **Product Information**—Lists all information that appears on the **Product View** for a package.
 - **Conflict Information**—Lists all conflicts found during the most recent **Conflict Wizard** execution, or the most recent import (if conflict identification was performed). These conflicts are also listed on the **Conflicts View** for that package.
 - **Validation Information**—Lists all validation errors and warnings found during the most recent validation execution for that package. These errors and warnings are also listed on the **Validation View** for that package.
 - **Extended Attributes**—Lists all user-defined information that appears on the **Extended Attributes View** for a package.
4. Click the section title to expand or contract that section of the report.
5. To print a report, click the **Print Page** link at the top of the report.

Generating File and Registry Reports

The Files Report lists of all of the files included with the package, and the Registry Report lists any registry entries created or changed by the package.



Task: *To generate a Files or Registry report :*

1. In the **Products View**, select the root node of the product that you want to generate a report for.
2. Click **File** or **Registry** on the **Reports** menu.

or

Click **File** or **Registry** under **Reports** on the context menu.

You are prompted to select a location where you want to save the report.

3. Select a location and click **Save**. The report opens in a browser window.

For the Files Report, the following information is displayed:

- **Product Information**—Lists all information that appears on the Product View for a package.
- **Files**—Lists all of the files that are included with the package. These files are also listed on the **Files/Components View** for that package.

For the Registry Report, the following information is displayed:

- **Product Information**—Lists all information that appears on the **Product View** for a package.
- **Registry**—Lists any registry entries created or changed by the package. These registry entries are also listed on the **Registry View** for that package.

4. Click the section title to expand or contract that section of the report.
5. To print a report, click the **Print Page** link at the top of the report.

Generating Reports in Crystal Reports Format

Three Crystal Reports are shipped with AdminStudio:

- **Conflicts**—Provides a report of all conflicts found during the most recent Conflict Wizard execution, or the most recent Import (if conflict identification was performed). This report is also available on a per-product basis by right-clicking on a product in the Product View.
- **Files**—Provides a comprehensive listing of all files in current Application Catalog, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View.
- **Registry**—Provides a comprehensive listing of all registry entries in the current Application Catalog, sorted on a per package and per component basis. This report is also available on a per-product basis by right-clicking on a product in the Product View.



Task: *To generate a Crystal Reports report:*

1. In the **Products View**, select the root node of the product that you want to generate a report for.
2. Select **Crystal Reports Conflicts**, **Crystal Reports Files**, or **Crystal Reports Registry** from the submenu.

The three Crystal Reports selections will be listed on the **Report** menu if the **Display RPT Reports** option on the **General** tab of the ConflictSolver **Options** dialog box is selected.

3. To print a report, click the **Print Page** link at the top of the report.

Creating Custom Reports

ConflictSolver stores imported packages in a relational database. Crystal Reports can be used to quickly query this database and generate reports.

The following procedure explains how to create a custom report for ConflictSolver using AdminStudio and Crystal Reports 8. The resulting report allows you to search ConflictSolver and generate reports based on the presence of a specified INI file. You must have Crystal Reports 8 installed for this procedure.

This procedure contains the following steps:

- [Step 1: Creating an ODBC File DSN](#)
- [Step 2: Creating the Base Report](#)
- [Step 3: Configuring Basic Settings](#)
- [Step 4: Creating Page Column Headings](#)
- [Step 5: Adding Database Fields](#)
- [Step 6: Adding Parameters](#)
- [Step 7: Creating Conditions for Filtering](#)
- [Step 8: Modifying the Footer](#)
- [Step 9: Using the Report in ConflictSolver](#)

Step 1: Creating an ODBC File DSN



Task: *To create an ODBC File DSN:*

1. From the **Windows Start** menu, point to **Programs**, then **Administrative Tools**, then click on **Data Sources (ODBC)**. This starts the Microsoft ODBC Data Source Administrator.
2. Click on the **File DSN** tab and click **Add**.
3. Select **Microsoft Access Driver (*.mdb)** from the drivers list and click **Next**.
4. Name the data source as **CSIniFileSearch** and click **Next**.
5. Click **Finish**.
6. Click **Select** and browse and select the ConflictSolver sample Access Application Catalog.
7. Click **OK**.
8. Click **OK** again.

Step 2: Creating the Base Report



Task: *To create the base report:*

1. Start Crystal Reports.
2. Select **Blank Report** either from the **Welcome** dialog box or from the **New Report** dialog box.
3. Expand **CSIniFileSearch.dsn** under **ODBC** from the **Data Explorer** dialog box.
4. Select **IniFile** from the list of tables under **CSIniFileSearch.dsn** and click **Add**.
5. Select **Component** from the list of tables under **CSIniFileSearch.dsn** and click **Add**.
6. Select **Package** from the list of tables under **CSIniFileSearch.dsn** and click **Add**.
7. Click **Close**.
8. Crystal Reports will use its smart linking feature to link the selected tables. Delete the link between **IniFile** and **Package**.
9. Create a link between **Component** and **Package** table by dragging the **PackageCode** field from the **Component** table and dropping into the **PackageCode** field in the **Package** table.
10. Click **OK** on the **Visual Linking Expert** dialog box.
11. From the **Field Explorer** dialog box, click **Close**. You should now see a blank report in the design mode with header, details and footer sections.

Step 3: Configuring Basic Settings



Task: *To configure basic settings:*

1. From the **File** menu, select **Printer Setup** and change the orientation from **Portrait** to **Landscape**.
2. From the **File** menu, select **Summary Info**.
3. Type **IniFile Search Report** in the **Title** text box, and click **OK**.
4. From the **Insert** menu, select **Special Field**.
5. Scroll down and select **Report Title** from the list and click on the **Insert to Report** toolbar button.
6. Click **Close**.

Table Name Prefixes for SQL Server Databases

If you are building a report from an SQL Server database, the names of the added tables may be prefixed by [database_name] . [user_name].

Delete these prefixes and just keep the table name, so that your report works even if you connect to other Application Catalogs.



Task: *To delete prefixes:*

1. Click on **Database | Set Location**.
2. Click on each table.
3. Look at the Table text box name under the **Location** frame and make sure that the table contains just the table names.

Step 4: Creating Page Column Headings



Task: *To create page column headings:*

1. Select **Text Object** from the Insert menu and drop the mouse on the left side inside the **Page Header** section.
2. Type **Product** in the text object.
3. Click on the above text object, press Ctrl+C to copy it, press Ctrl+V and drop the mouse adjacent to the first text object. Change the text from **Product** to **Component**.
4. Repeat the above step and add the columns named File Name, Path, INI Section, INI Key, INI Key Value, and Action.
5. If desired, you can right-click on the text objects and select **Change Font** to update the font.

Step 5: Adding Database Fields



Task: *To add database fields:*

1. Select **Database Fields** from the **Insert** menu.
2. Expand the **Package** table node, press **Enter** on the **ProductName** field, and drop the mouse in the **Details** section under the **Product** page heading.
3. Expand the **Component** table, select **Component** field, press **Enter**, and drop the mouse in the **Details** section under the **Component** page heading.
4. Select the **IniFile** table and drop **FileName**, **FullPath**, **IniSection**, **IniKey**, **IniValue**, and **IniAction** fields.
5. Select all the fields in the detail section
6. Right-click on the fields and select **Format Object**.
7. Check the **Can Grow** box.

8. Save the report file under the following folder:

AdminStudio Installation Directory\ConflictSolver\ Reports\Standard\MS

Step 6: Adding Parameters

Standard ConflictSolver reports require at least two parameters to be defined in the report file.



Task: *To add parameters:*

1. From the **Insert** menu, select **Parameter** Field.
2. Click on **New** from the toolbar or press Ctrl+N.
3. Type the parameter name as **ProductName** and leave the type to default value of **String**.
4. Repeat this step and add one more parameter named **ProductVersion** of type **String**.
5. Click **Close**.

ConflictSolver automatically passes values for these parameters. In ConflictSolver when the report is selected from the main menu item, these parameters are passed as *; and when the report is selected by right clicking on the package, ConflictSolver passes the ProductName and ProductVersion for the selected package.

For this report, a prompt is needed so you can enter an INI file name, and based on that, show the filtered record. This requires another parameter.



Task: *To add the third parameter:*

1. From the Insert menu, select **Parameter Field**.
2. Type the parameter name as **INIFileName**.
3. Set the type to **String**.
4. Type the following into the **Prompting Text** text box:
Enter the INI File Name. Use * as a wildcard character.
5. Click **OK**.
6. Click **Close** to dismiss the **Field Explorer** dialog box.

Step 7: Creating Conditions for Filtering



Task: *To create the condition to filter the records based on the input parameters (ProductName, ProductVersion, and INIFileName):*

1. From the **Report** menu, point to **Edit Selection Formula** and select **Record**.

2. Type the following text in the **Filter Criteria** text box:

```
{IniFile.FileName} like {?INIFileName}andif {?ProductName} = '*' then TRUEelse  
  {Package.ProductName} = {?ProductName}andif {?ProductVersion} = '*' then TRUEelse  
  {Package.ProductVersion} = {?ProductVersion}
```

Crystal Reports uses the above selection formula while sending the SQL query to the back end database server. The above formula makes the WHERE clause in the SELECT database query, which is internally sent by Crystal Reports while viewing the report. The above formula also filters the Application Catalog records based on the IniFile.FileName, and if ProductName/ProductVersion are passed they are used, else only the IniFile.FileName is used in the selection criteria.

3. Click **Save** from the toolbar.

Step 8: Modifying the Footer



Task: *To add the date/time and page number to the report footer:*

1. From the **Insert** menu, select **Special Field**.
2. Select **Page N of M** and drop it on the report within the **Report Footer** section.
3. Repeat for **Print Date** and **Print Time**.
4. Save the report.
5. Close Crystal Reports and start ConflictSolver.

Step 9: Using the Report in ConflictSolver

The Reports menu item in ConflictSolver is dynamically generated based on the .rpt report files present in the Reports folder. Start ConflictSolver, open the sample Application Catalog, click on Options from the Tools menu, and ensure that the above report is saved in the Reports folder as specified in the General tab.

If you select INIFile Search Report from the Reports menu, you can access the report you just created.



Task: *To use the report:*

1. From the **Reports** menu, point to **Standard** and then select **IniFile Search Report**.
2. Type **system.ini** in the text box and click **OK**. A report is generated. You can use ***** as the wildcard character while typing the INI file name.

Open some other Application Catalog for ConflictSolver, run the report, type the name of the file and verify the results by looking at the report. You can also right-click on the package in ConflictSolver and generate the IniFile Search report only for the selected package.

Application Conflict Evaluators (ACEs)

Application Conflict Evaluators (ACEs) are the rules that allow ConflictSolver to identify conflicts between Windows Installer-based setups, and between Windows Installer-based setups and operating system images (OS Snapshots). ConflictSolver can automatically solve some of the conflicts spelled out in these rules. When an automatic remedy is available, the associated rule used to fix the conflict is spelled out in a Conflict Application Resolution Definition (CARD).

This table lists each of the Best Practice ACE rules, and is grouped by conflict type:

Table 13-10 • Best Practice ACE Rules

Conflict Type	Associated Rules
Components	ACE04 , ACE05 , ACE06 ,
Merge Module Integrity	ACE26 , ACE36
Recommended Rules	ACE25 , ACE27 , ACE28 , ACE29 , ACE31 , ACE32 , ACE33 , ACE34 , ACE35
Windows Terminal Server Compatibility	WTS01 , WTS02 , WTS03 , WTS04 , WTS05

This table lists each of the Conflict ACE rules, and is grouped by conflict type.

Table 13-11 • Conflict ACE Rules

Conflict Type	Associated Rules
Component	ACE02 , ACE09 , ACE30
File Extensions	ACE17
Files	ACE03 , ACE07 , ACE08 , ACE12 , ACE23

Table 13-11 • Conflict ACE Rules

Conflict Type	Associated Rules
INI Files	ACE14 , ACE21 , ACE22
NT Services	ACE16
ODBC Resources	ACE15
Product Properties	ACE18 , ACE19 , ACE20
Registry	ACE10 , ACE24
Shortcuts	ACE13

User-Defined ACEs

In addition to these ACEs included with ConflictSolver, you can also create three types of user-defined ACEs to use when detecting conflicts: Source Only, Source and Target, and DLL-Based. See [User-Defined ACEs](#) for more information.

ACE Index

The following list provides links to each individual ACE. These ACEs are used to identify potential conflicts between installation packages.

Table 13-12 • ACE Index

ACE Rule	Rule Type	Conflict Type	Brief Description of ACE
ACE02	Conflict	Component	Checks to see if components in different packages that have matching ComponentIds also have identical destination paths.
ACE03	Conflict	Files	Checks to see if components in different packages that have matching ComponentIds also contain the same files.
ACE04	Best Practice	Component	Checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the CreateFolder table.
ACE05	Best Practice	Component	Checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package.
ACE06	Best Practice	Component	Checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file.

Table 13-12 • ACE Index (cont.)

ACE Rule	Rule Type	Conflict Type	Brief Description of ACE
ACE07	Conflict	Files	Checks for the existence of the same file in components with different ComponentIds.
ACE08	Conflict	Files	Identifies components with identical ComponentIds, and checks those components to see if the versions of the files in each component match.
ACE09	Conflict	Component	Checks to see if merge modules with identical ComponentIds are identical.
ACE10	Conflict	Registry	Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.
ACE12	Conflict	Files	Checks for components that contain files that can be replaced by one of the merge modules that you have imported into the Application Catalog database. (Before running ACE12, you should import all merge modules you are likely to use at your organization.)
ACE13	Conflict	Shortcuts	Checks for the existence of the same shortcut within different packages in components with different ComponentIds.
ACE14	Conflict	INI Files	Checks for the existence of components with different ComponentIds that modify the same INI file entry, such as the [File Name/Section/Key/Value] entry.
ACE15	Conflict	ODBC Resources	Checks for the existence of identical ODBC entries in components with different ComponentIds.
ACE16	Conflict	NT Services	Checks for the existence of identical NT Services in components with different ComponentIds.
ACE17	Conflict	File Extensions	Checks for identical file Extension/Verb combinations in components with different ComponentIds.
ACE18	Conflict	Product Codes	Checks the Package Code to see if it is unique.
ACE19	Conflict	Product Codes	Checks the Product Code to see if it is unique.
ACE20	Conflict	Product Codes	Checks the Upgrade Code to see if it is unique.

Table 13-12 • ACE Index (cont.)

ACE Rule	Rule Type	Conflict Type	Brief Description of ACE
ACE21	Conflict	INI Files	Checks entries in the IniFile table to see if they conflict with similar entries in the File table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated.
ACE22	Conflict	INI Files	Checks file name/target directory pairs in the File table to see if they conflict with similar entries in the IniFile table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated.
ACE23	Conflict	Files	Identifies file duplication between source and target packages. ACE23 checks to see if files with the same name and destination directory have the same size, version, and language when comparing a source package against a target package. If a file with the same name and destination directory is found in both the source and target packages, but the file has a different size, version, or language, ACE23 fails.
ACE24	Conflict	Registry	Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. If a registry entry with the same registry hive, key, and value name is found in both the source and target package, but the registry entry has a different data type or value, ACE24 fails.
ACE25	Best Practice	Recommended Rules	A check of the CustomAction table to identify any hard coded paths.
ACE26	Best Practice	Merge Module Integrity	A check of the Merge Modules in a package to confirm that they also exist in the Application Catalog.
ACE27	Best Practice	Recommended Rules	Checks to see if data exists in the DuplicateFile MSI table that is not being executed with an associated DuplicateFiles action.
ACE28	Best Practice	Recommended Rules	A check of the Environment table to identify any hard coded paths.
ACE29	Best Practice	Recommended Rules	A check of the IniFile table to identify any hard coded paths.
ACE30	Conflict	Components	A rule to identify components with different ComponentIDs installing the same key file to the same directory. (A check for KeyPath conflicts across components.)

Table 13-12 • ACE Index (cont.)

ACE Rule	Rule Type	Conflict Type	Brief Description of ACE
ACE31	Best Practice	Recommended Rules	Checks to see if data exists in the MoveFile MSI table that is not being executed with an associated MoveFiles action.
ACE32	Best Practice	Recommended Rules	A check of the Registry table to identify any hard coded paths.
ACE33	Best Practice	Recommended Rules	Checks to see if data exists in the RemoveFile MSI table that is not being executed with an associated RemoveFiles action.
ACE34	Best Practice	Recommended Rules	Checks to see if data exists in the RemoveIniFile MSI table that is not being executed with an associated RemoveIniFiles action.
ACE35	Best Practice	Recommended Rules	Checks to see if data exists in the RemoveRegistry MSI table that is not being executed with an associated RemoveRegistryValues action.
ACE36	Best Practice	Merge Module Integrity	A check of the required Merge Modules in a package to confirm that they also exist in the Application Catalog.
WTS01	Best Practice	Windows Terminal Server Compatibility	Checks to see if the ALLUSERS property is defined.
WTS02	Best Practice	Windows Terminal Server Compatibility	Checks for any user specific registry entries.
WTS03	Best Practice	Windows Terminal Server Compatibility	Checks for any per-user component destination.
WTS04	Best Practice	Windows Terminal Server Compatibility	Checks for any per-user ODBC data source.
WTS05	Best Practice	Windows Terminal Server Compatibility	Checks for any per-user environment settings.

ACE02

ACE02 checks to see if components in different packages that have matching ComponentIds also have identical destination paths. The ACE02 Error String is displayed in the following format:

The destination [PATH1] for the component [COMPONENT1] in the package [PACKAGE1] conflicts with the destination for the component [COMPONENT1] in the package [PACKAGE2]. The correct destination should be [PATH2].

Summary

Table 13-13 • ACE02 Summary

Topic	Description
Conflict Type:	Conflict Rule / Components
Description:	Checks to see if components in different packages that have matching ComponentIds also have identical destination paths.
Result:	If components with the same ComponentId have different destination paths, ACE02 fails.
Resolution Type:	Automatic (CARD02)
Resolution:	The destination path of the component in the Source package is automatically set to match that of the component in the Target package.

ACE03

ACE03 checks to see if components in different packages that have matching ComponentIds also contain the same files. The ACE03 Error String is displayed in the following format:

The file(s) [FILENAME] in the component [COMPONENT1] in the package [PACKAGE1] are either new to or missing from the component [COMPONENT1] in the package [PACKAGE2].

Summary

Table 13-14 • ACE03 Summary

Topic	Description
Conflict Type:	Conflict Rule / Files
Description:	Checks to see if components in different packages that have matching ComponentIds also contain the same files.
Result:	If components with the same ComponentId do not contain the same files (either files are missing or they are different versions), ACE03 fails.

Table 13-14 • ACE03 Summary

Topic	Description
Resolution Type:	Manual
Resolution:	Use InstallShield Editor to create a Windows Installer Transform (.mst) file for the Source package MSI so that the component in the Source package contains the same files as the component in the Target package.

Cause

The ComponentIds for components in two different MSI packages are identical but either files are missing or different in those components.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task:

To resolve this conflict:

1. Open the MSI package in InstallShield Editor.
2. Add or modify the components as necessary so that the file(s) is present in both components and the versions match in both MSI packages. To accomplish this:
 - a. Open the **Components View** and find the component that needs to be modified.
 - b. Expand the list under the component and select **Files**. A list of the files included with that component is displayed.
 - c. To delete a file that is not present in the other component, select the file, right-click and choose **Delete** from the menu.
 - d. If you need to add a new file to match the other component, right click anywhere in the **Files** list, choose **Add** from the menu, and add the correct file.
3. Select **Save As** from the **File** menu and save the changes as a Windows Installer Transform (.mst) file.
4. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE03 again.

ACE04

ACE04 checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the CreateFolder table. The ACE04 Error String is displayed in the following format:

The component [COMPONENT1] in the package [PACKAGE1] does not have a key file. Additionally, the directory/component pair is not listed in the CreateFolder table.

Summary

Table 13-15 • ACE04 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Components
Description:	Checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the CreateFolder table.
Result:	If a component does not have a key file, and if the directory/component pair is not listed in the CreateFolder table, ACE04 fails.
Resolution Type:	Automatic (CARD04)
Resolution:	A CreateFolder entry is created for the component.

ACE05

ACE05 checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package. The ACE05 Error String is displayed in the following format:

The component [COMPONENT1] in the package [PACKAGE1] has more than one executable module (*.chm, *.dll, *.drv, *.exe, *.hlp, *.ocx, *.sys, or *.tlb). The file [EXECUTABLE_MODULE_FILENAME] must be included as a separate component.

Summary

Table 13-16 • ACE05 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Components
Description:	Checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package.
Result:	If more than one executable exists in a component, ACE05 fails.
Resolution Type:	Automatic (CARD05)
Resolution:	Modifies the component so that only one EXE or DLL exists, and it adds new components for remaining EXE, DLL, OCX, HLP, CHM, TLB, SYS, and DRV files.

ACE06

ACE06 checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file. The ACE06 Error String is displayed in the following format:

The component [COMPONENT1] in the package [PACKAGE1] does not have an executable module (*.chm, *.dll, *.drv, *.exe, *.hlp, *.ocx, *.sys, or *.tlb) as the key file. The current key file [NON-EXECUTABLE_FILENAME] can be replaced with [EXECUTABLE_MODULE_FILENAME].

Summary

Table 13-17 • ACE06 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Components
Description:	Checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file.
Result:	If the executable module is not the key file, ACE06 fails.
Resolution Type:	Automatic (CARD06)
Resolution:	The executable module is automatically made the key file.

ACE07

ACE07 checks for the existence of the same file in components with different ComponentIds. ACE07 reports four types of errors, depending on the various combinations of source and target files and whether or not they originated in a merge module:

Table 13-18 • Four Types of ACE07 Errors

Source File	Target File	Severity	CARD-Enabled	Conflict View Message	Output Message
Not Merge Module	Merge Module	Warning	No	The file [FILENAME] in the component [COMPONENTNAME] is identical to the merge module installed file in the component [COMPONENTNAME]. Confirm this error by running this package against ACE12.	The file [FILENAME] in the component [COMPONENTNAME] in the package [PACKAGENAME] is identical to the merge module installed file in the component [COMPONENTNAME] in the package [PACKAGENAME]. Confirm this error by running this package against ACE12.

Table 13-18 • Four Types of ACE07 Errors

Source File	Target File	Severity	CARD-Enabled	Conflict View Message	Output Message
Not Merge Module	Not Merge Module	Error	Yes	The file [FILENAME] in the component [COMPONENTNAME] is identical to a file in the component [COMPONENTNAME], but the components have different GUIDs.	The file [FILENAME] is identical in both the component [COMPONENTNAME] in the package [PACKAGENAME] and the component [COMPONENTNAME] in the package [PACKAGENAME], but the components have different GUIDs.
Merge Module	Merge Module	Error	No	The file [FILENAME] in the merge module installed component [COMPONENTNAME] is identical to a file in another merge module installed component [COMPONENTNAME], but the components have different GUIDs. Run ACE12 to determine which merge module is most appropriate to use.	The file [FILENAME] is identical and installed by merge modules in both the component [COMPONENTNAME] in the package [PACKAGENAME] and the component [COMPONENTNAME] in the package [PACKAGENAME], but the components have different GUIDs. Run ACE12 to determine which merge module is most appropriate to use.
Merge Module	Not Merge Module	Error	No	The file [FILENAME] in the merge module installed component [COMPONENTNAME] is identical to the file in the component [COMPONENTNAME]. Confirm this error by running the [PACKAGENAME] package against ACE12.	The file [FILENAME] in the merge module installed component [COMPONENTNAME] in the package %s is identical to the file in the component [COMPONENTNAME] in the package [PACKAGENAME]. Confirm this error by running the [PACKAGENAME] package against the ACE12.

Summary

Table 13-19 • ACE07 Summary

Topic	Description
Conflict Type:	Conflict Rule / Files
Description:	Checks for the existence of the same file in components with different ComponentIds.

Table 13-19 • ACE07 Summary

Topic	Description
Result:	ACE07 reports four types of errors, depending on the various combinations of source and target files and whether or not they originated in a merge module. (See table above.)
Resolution Type:	Automatic (CARD07) only if both the source and target files originated in a Merge Module. For all other errors, confirm the error by running the package against ACE12 and then, based upon the results, decide how to proceed.
Resolution:	The Source package ComponentId is changed to match the Target package Component Id.

ACE08

ACE08 identifies components with identical ComponentIds, and checks those components to see if the versions of the files in each component match. The ACE08 Error String is displayed in the following format:

The file [FILENAME], version [VERSION_NUMBER1], in the component [COMPONENT1] conflicts with the same file, version [VERSION_NUMBER2], in the component [COMPONENT2].



Note • ACE08 is a warning, not an error.

Summary

Table 13-20 • ACE08 Summary

Topic	Description
Conflict Type:	Conflict Rule / Files
Description:	Identifies components with identical ComponentIds, and checks those components to see if the versions of the files in each component match.
Result:	If components with identical ComponentIds contain files with different versions, ACE08 fails.
Resolution Type:	Manual
Resolution:	Change the file versions to match those of the Target component, or change the file versions in the Source component.

Cause

Two components with identical ComponentIds in different packages have the same file with different versions.

Resolution

This conflict requires a manual resolution. Follow these steps:

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, there are two options that can be used to resolve the issue:

Option #1: Change the Component Code. To accomplish this:

- a. Open the **Components View** and select the component that needs to be modified.
- b. In the Component's **Property** grid, change the value in the **Component Code** field.

Option #2: Replace the files so that the versions match in both MSI packages. To accomplish this:

- a. Open the **Components View** and find the component that needs to be modified.
 - a. Expand the listing under the component, and select **Files**. The list of files included with that component is displayed.
 - b. To delete a file with the wrong version, select the file, right-click and choose **Delete** from the menu.
 - c. To add the correct version of a file, right click anywhere in the **Files** list, choose **Add** from the menu, and add the new file with the correct version.
3. Select **Save As** on the **File** menu and save the changes as a Windows Installer Transform (.mst) file.
 4. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE08 again.

ACE09

ACE09 checks to see if merge modules with identical ComponentIds are identical. The ACE09 Error String is displayed in the following format:

The merge module, [MERGE_MODULE_NAME], version [VERSION_NUMBER1], conflicts with the same merge module, version [VERSION_NUMBER2], in another package.



Note • ACE09 is a warning, not an error.

Summary

Table 13-21 • ACE09 Summary

Topic	Description
Conflict Type:	Conflict Rule / Components

Table 13-21 • ACE09 Summary

Topic	Description
Description:	Checks to see if merge modules with identical ComponentIds are identical.
Result:	If merge modules with identical ComponentIds are different, a warning is generated.
Resolution Type:	Manual
Resolution:	Obtain the latest version of the merge module and rebuild the MSI.

Cause

Two merge modules with identical ComponentIds but different versions are found in different packages.

Resolution



Task:

To perform a manual resolution:

1. Obtain the latest version of the merge module displayed in the ACE message. The merge module may be available for download at the vendor's web site.
2. Open the MSI package in **InstallShield Editor**.
3. Place the new merge module in one of the Merge Module Locations that are specified on the **Merge Modules** tab of the ConflictSolver **Options** dialog box. This allows the merge module to be displayed in the **Redistributables** view of the InstallShield Editor.
4. Navigate to the **Redistributables View** in the InstallShield Editor project. Deselect the old merge module causing the conflict and select the new merge module in its place.
5. Select **Save As** on the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
6. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE09 again.

ACE10

ACE10 checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds. The ACE10 Error String is displayed in the following format:

The registry entry [REGISTRY_ENTRY] [REGISTRY_KEY] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same registry entry in the component [COMPONENT1] in the package [PACKAGE2].



Note • ACE10 is a warning, not an error.

Summary

Table 13-22 • ACE10 Summary

Topic	Description
Conflict Type:	Conflict Rule / Registry
Description:	Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.
Result:	If the same Root/Key/Name registry combination is in more than one component, a warning is generated.
Resolution Type:	Manual
Resolution:	Change the ComponentId of the Source component to match that of the Target component.

Cause

The same Root/Key/Name registry combination is found in more than one component.

Conditions When an ACE10 Error Can Be Ignored

ACE10 uses data from the MSI Registry table to check for identical registry Root/Key/Name combinations in different components. However, there may be situations in which ACE10 will report an error unnecessarily. MSI supports a grammar for the Registry table in which if the Value field is preceded or terminated by the sequence tilde '[~]', then the registry value will be appended or prepended, respectively, to the existing registry Value. This sort of operation may be perfectly acceptable if the applications in question are modifying a common registry key in a manner consistent with its purpose.

You will need to decide individually if an ACE10 error is valid, but consider checking the Registry Value field, as its contents may prove useful in helping you decide.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View** and find the component that needs to be modified by referring to the component name displayed in the error message.
3. In the Component's **Property** grid, change the value in the **Component Code** field to match the Component Code (ComponentId) of the component in the other project.

If it is unclear what the value should be changed to, follow these steps:

- a. Open up the package that will not be modified in Direct Edit mode by selecting Direct Editor under **Additional Tools**. A list of the package tables is displayed.
- b. Select the **Component** table and search for the **Component Name** displayed in the conflict.
- c. Once the entry is found, take note of the value displayed in the **ComponentId** column. This is the Component Code that should be used for the component in the other package.
4. Select **Save As** from the **File** menu and save the changes as a Windows Installer Transform (.mst) file.
5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE10 again.

ACE12

ACE12 checks for components that contain files that can be replaced by one of the merge modules that you have imported into the Application Catalog database. (Before running ACE12, you should import all merge modules you are likely to use at your organization into the Application Catalog database.)

ACE12 reports an Error if the file in question originated from a merge module and the merge module was found in the Application Catalog. If the merge module was not found in the Application Catalog, ACE12 reports a Warning:

Table 13-23 • ACE12 Output Summary

Package File	Severity	Output
Not from Merge Module	Error	The file [FILENAME] in the component [COMPONENT1] in the package [PACKAGE1] should be replaced with the merge module [MERGE_MODULE_NAME].
From Merge Module	Warning	The [FILENAME] file originating from the [ModuleID] Merge Module in package [PACKAGE1] is a candidate to be replaced with the [MERGE_MODULE_NAME]. However, the [ModuleID] Merge Module is not in the Application Catalog which makes proper evaluation impossible.

Summary

Table 13-24 • ACE12 Summary

Topic	Description
Conflict Type:	Conflict Rule / Files
Description:	Checks for components that contain files that can be replaced by one of the merge modules that you have imported into the Application Catalog database. (Before running ACE12, you should import all merge modules you are likely to use at your organization.)

Table 13-24 • ACE12 Summary

Topic	Description
Result:	ACE12 reports a Warning if the file in question originated from a merge module and the merge module was found in the Application Catalog. If the merge module was not found in the Application Catalog, ACE12 reports an Error.
Resolution Type:	Manual
Resolution:	Rebuild the package using the merge module rather than the duplicate files.

Cause

A component within one package contains a file that is found in a merge module of another package. Using merge modules is always preferable as a way to install files in a consistent fashion.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task:

To resolve this conflict:

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View**, and find the component that needs to be modified by referring to the component name displayed in the error message.
3. Expand the list under the component and select **Files**. A list of the files included with that component is displayed.
4. Select the file that was displayed in the error message, right-click, and select **Delete** from the menu.
5. Take note of the feature that the component is associated with and then navigate to the **Redistributables View**.
6. In the **Redistributables View**, select the appropriate merge module.
7. After selecting the merge module, move to the **Conditional Installation View** on the right side and select the feature that was mentioned earlier.
8. Rebuild the package.
9. Open ConflictSolver and reimport this package into your Application Catalog, and then use the Conflict Wizard to check it against ACE12 again.

ACE13

ACE13 checks for the existence of the same shortcut within different packages in components with different ComponentIds. The ACE13 Error String is displayed in the following format:

The shortcut [SHORTCUT_NAME] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same shortcut in the component [COMPONENT2] in the package [PACKAGE2].

Summary

Table 13-25 • ACE13 Summary

Topic	Description
Conflict Type:	Conflict Rule / Shortcuts
Description:	Checks for the existence of the same shortcut within different packages in components with different ComponentIds.
Result:	If the same shortcut exists in more than one component, ACE13 fails.
Resolution Type:	Manual
Resolution:	Change the ComponentId of the Source component to match the Component Id of the Target component.

Cause

The same shortcut is found in components with different ComponentIDs in two different packages.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View**, and find the component that needs to be modified by referring to the component name displayed in the error message.
3. In the Component's **Property** grid, change the value in the **Component Code** field to match the ComponentId of the component in the other project.

If it is unclear what the value should be changed to, follow these steps:

- a. Open up the package that will not be modified in Direct Edit mode by selecting **Open** on the **File** menu, and then selecting **Direct Editor** under **Additional Tools**. A list of the package tables is displayed.
- b. Select the **Component** table and search for the **Component Name** displayed in the conflict.

- c. Once the entry is found, take note of the value displayed under the **ComponentId** column. This is the **Component Code** that should be used for the component in the other package.
4. Select **Save As** on the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE13 again.

ACE14

ACE14 checks for the existence of components with different ComponentIds that modify the same INI file entry, such as the [File Name/Section/Key/Value] entry. If this condition is found, ACE14 returns the following Warning:

The INI file entry [INI_FILE_ENTRY] in the file [INI_FILE_NAME] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same INI file entry in the component [COMPONENT1] in the package [PACKAGE2].

Summary

Table 13-26 • ACE14 Summary

Topic	Description
Conflict Type:	Conflict Rule / INI Files
Description:	Checks for the existence of components with different ComponentIds that modify the same INI file entry, such as [File Name/Section/Key/Value].
Result:	If the same INI file entry is modified by different components, ACE14 fails.
Resolution Type:	Manual using the instructions under Resolution below
Resolution:	Change the ComponentId of the Source component to match the ComponentId of the Target component.

Cause

The same INI file is found in components with different ComponentIds in two different packages.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task:

To resolve this conflict:

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View**, and find the component that needs to be modified by referring to the component name displayed in the Warning.

3. In the Component's **Property** grid, change the value in the **Component Code** field to match the Component Code (ComponentId) of the component in the other project.

If it is unclear what the value should be changed to, follow these steps:

- a. Open up the package that will not be modified in Direct Edit mode by selecting **Open** on the **File** menu, and then selecting **Direct Editor** under **Additional Tools**. A list of the package tables is displayed.
 - b. Select the **Component** table and search for the **Component Name** displayed in the conflict.
 - c. Once the entry is found, take note of the value displayed under the **ComponentId** column. This is the Component Code that should be used for the component in the other package.
4. Select **Save As** on the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
 5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE14 again.

ACE15

ACE15 checks for the existence of identical ODBC entries in components with different ComponentIds. The ACE15 Error String is displayed in the following format:

The ODBC entry [ODBC_ENTRY] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same ODBC entry in the component [COMPONENT1] in the package [PACKAGE2].

Summary

Table 13-27 • ACE15 Summary

Topic	Description
Conflict Type:	Conflict Rule / ODBC Resources
Description:	Checks for the existence of identical ODBC entries in components with different ComponentIds.
Result:	If identical ODBC entries exist in components with different ComponentIds, ACE15 fails.
Resolution Type:	Automatic (CARD15)
Resolution:	ConflictSolver changes the Source ComponentId to match the Target ComponentId.

ACE16

ACE16 checks for the existence of identical NT Services in components with different ComponentIds. The ACE16 Error String is displayed in the following format:

The NT service [SERVICE1] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same NT Service in the component [COMPONENT1] in the package [PACKAGE2].

Summary

Table 13-28 • ACE16 Summary

Topic	Description
Conflict Type:	Conflict Rule / NT Services
Description:	Checks for the existence of identical NT Services in components with different ComponentIds.
Result:	If identical NT Services exist within different components, ACE16 fails.
Resolution Type:	Manual
Resolution:	Change the ComponentId of the Source component to match the ComponentId of the Target component.

Cause

The same NT Service is found in components with different ComponentIDs in two different packages.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open the package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View**, and find the component that needs to be modified by referring to the component name displayed in the error message.
3. In the Component's **Property** grid, change the value in the **Component Code** field to match the Component Code (ComponentId) of the component in the other project.

If it is unclear what the Component Code should be changed to, follow these steps:

- a. Open up the package that will not be modified in Direct Edit mode by selecting **Open** on the **File** menu, and then selecting **Direct Editor** under **Additional Tools**. A list of the package tables is displayed.
- b. Select the **Component** table and search for the **Component Name** displayed in the conflict.

- c. Once the entry is found, take note of the value displayed in the **ComponentId** column. This is the Component Code that should be used for the component in the other package.
4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE16 again.

ACE17

ACE17 checks for identical file Extension/Verb combinations in components with different ComponentIds. The ACE17 Error String is displayed in the following format:

The verb [VERB_NAME] in extension [EXTENSION_NAME] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same verb & extension in the component [COMPONENT1] in the package [PACKAGE2].

Summary

Table 13-29 • ACE17 Summary

Topic	Description
Conflict Type:	Conflict Rule / File Extensions
Description:	Checks for identical file Extension/Verb combinations in components with different ComponentIds.
Result:	If identical file Extension/Verb combinations exist in components with different ComponentIds, ACE17 fails.
Resolution Type:	Manual
Resolution:	Change the ComponentId of the Source component to match the ComponentId of the Target component.

Cause

The same Extension/Verb combination is found in components with different ComponentIds in two different packages.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open the transform file or MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components View**, and find the component that needs to be modified by referring to the component name displayed in the error message.
3. In the Component's **Property** grid, change the value in the **Component Code** field to match the Component Code (ComponentId) of the component in the other project.

If it is unclear what the **Component Code** should be changed to, follow these steps:
 - a. Open up the package that will not be modified in Direct Edit mode by selecting **Open** on the **File** menu, and then selecting **Direct Editor** under **Additional Tools**. A list of the package tables is displayed.
 - b. Select the **Component** table and search for the Component Name displayed in the conflict.
 - c. Once the entry is found, take note of the value displayed in the **ComponentId** column. This is the **Component Code** that should be used for the component in the other package.
4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE17 again.

ACE18

ACE18 checks the Package Code to see if it is unique. The ACE18 Error String is displayed in the following format:

The package code [PACKAGE_CODE] in the package [PACKAGE1] is the same as the package code in the package [PACKAGE2].

Summary

Table 13-30 • ACE18 Summary

Topic	Description
Conflict Type:	Conflict Rule / Product Properties
Description:	Checks the Package Code to see if it is unique.
Result:	If the Package Code is identical to any other Package Code in ConflictSolver, ACE18 fails.
Resolution Type:	Automatic (CARD18)
Resolution:	A new, unique Package Code is automatically generated to replace the duplicate code.



Note • ACE18 can only be resolved automatically when applying changes directly to the original Windows Installer package. It cannot be used when resolving using a transform.

ACE19

ACE19 checks the Product Code to see if it is unique. The ACE19 Error String is displayed in the following format:

The product code [PRODUCT_CODE] in the package [PACKAGE1] is the same as the product code in the package [PACKAGE2].

Summary

Table 13-31 • ACE19 Summary

Topic	Description
Conflict Type:	Conflict Rule / Product Properties
Description:	Checks the Product Code to see if it is unique.
Result:	If the Product Code is identical to any other Product Code in ConflictSolver, ACE19 fails.
Resolution Type:	Automatic (CARD19)
Resolution:	A new, unique Product Code is automatically generated to replace the duplicate code.

ACE20

ACE20 checks the Upgrade Code to see if it is unique. The ACE20 Error String is displayed in the following format:

The upgrade code [UPGRADE_CODE] in the package [PACKAGE1] is the same as the upgrade code in the package [PACKAGE2].

Summary

Table 13-32 • ACE20 Summary

Topic	Description
Conflict Type:	Conflict Rule / Product Properties
Description:	Checks the Upgrade Code to see if it is unique.
Result:	If the Upgrade Code is not unique, ACE20 fails.
Resolution Type:	Automatic (CARD20)

Table 13-32 • ACE20 Summary

Topic	Description
Resolution:	A new, unique Upgrade Code is automatically generated to replace the duplicate code.

ACE21

ACE21 checks entries in the IniFile table to see if they conflict with similar entries in the File table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated. The ACE21 Error String is displayed in the following format:

The INI file [INI_FILENAME] installed using the IniFile table to the destination [PATH_NAME] in the package [PACKAGE1.MSI] conflicts with the same file in the File table in the package [PACKAGE2].



Note • If performing conflict analysis with multiple source packages, and if the **Check conflicts across all source and target packages** option on the **Rules** tab of the **Options** dialog box is selected, both ACE21 and ACE22 are automatically evaluated when either one is selected. If only one of them is evaluated, then there is the possibility that conflicts will not be detected.

Summary

Table 13-33 • ACE21 Summary

Topic	Description
Conflict Type:	Conflict Rule / INI Files
Description:	Checks entries in the IniFile table to see if they conflict with similar entries in the File table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated.
Result:	If an entry in the IniFile table of the first MSI package duplicates a file name in a component of the File table of the second MSI package, and both are set to be installed to the same destination, ACE21 fails.
Resolution Type:	Manual
Resolution:	Ensure identically named INI files with identical destinations are identical in both the File and IniFile tables. This may involve adding or deleting sections or changing values in the INI files.

Cause

An entry in the IniFile table of the first MSI package duplicates a file name in a component in the File table of the second MSI package and both files are set to be installed to the same destination.

Resolution

This conflict requires a manual resolution.



Task: To perform a manual resolution:

1. In InstallShield Editor, open a transform file or MSI package that has an entry in the IniFile table.
2. Once the project is open, navigate to the **INI Files Changes View**, and find the INI file entry causing the conflict by referring to the INI file name displayed in the error message.
3. Analyze the sections, keywords, and values in this INI file and compare them to the INI file in the other MSI package. Edit the sections, keywords, and values in this INI file so that they match those of the second INI file. This may require adding and/or deleting data from the **INI File Changes View**.

In order to retrieve the INI file from the other MSI package, you may need to extract it from the MSI file itself or a cabinet file if the files are not uncompressed on the source media.

4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE21 again.

ACE22

ACE22 checks file name/target directory pairs in the File table to see if they conflict with similar entries in the IniFile table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated. The ACE22 Error String is displayed in the following format:

The INI file [INI_FILENAME] installed using the File table to the destination [PATH_NAME] in the package [PACKAGE3.MSI] conflicts with the same file in the IniFile table in the package [PACKAGE4].



Note • If performing conflict analysis with multiple source packages, and if the **Check conflicts across all source and target packages** option on the **Rules** tab of the **Options** dialog box is selected, both ACE21 and ACE22 are automatically evaluated when either one is selected. If only one of them is evaluated, then there is the possibility that conflicts will not be detected.

Summary

Table 13-34 • ACE22 Summary

Topic	Description
Conflict Type:	Conflict Rule / INI Files

Table 13-34 • ACE22 Summary

Topic	Description
Description:	Checks file name/target directory pairs in the File table to see if they conflict with similar entries in the IniFile table. The IniFile and File tables can change the same physical file. As a result, this ACE identifies these duplicate file changes so that they can be evaluated.
Result:	If a file name/target directory pair in a component in the File table is also listed as an IniFile table entry, ACE22 fails.
Resolution Type:	Manual
Resolution:	Ensure identically named INI files with identical destinations are identical in both the File and IniFile tables. This may involve adding or deleting sections or changing values in the INI files.

Cause

An INI file in a component of the first MSI package duplicates an entry in the IniFile table of the second MSI package and both files are set to be installed to the same destination.

Resolution

This conflict requires a manual resolution.



Task: *To perform a manual resolution:*

1. In InstallShield Editor, open a transform file or MSI package that has an entry in the IniFile table.
2. Once the project is open, navigate to the **INI Files Changes View**, and find the INI file entry causing the conflict by referring to the INI file name displayed in the error message.
3. Analyze the sections, keywords, and values in this INI file and compare them to the INI file in the other MSI package. Edit the sections, keywords, and values in this INI file so that they match those of the second INI file. This may require adding and/or deleting data from the **INI File Changes View**.

In order to retrieve the INI file from the other MSI package, you may need to extract it from the MSI file itself or a cabinet file if the files are not uncompressed on the source media.
4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE22 again.

ACE23

ACE23 identifies file duplication between source and target packages. ACE23 checks to see if files with the same name and destination directory have the same size, version, and language when comparing a source package against a target package. If a file with the same name and destination directory is found in both the source and target packages, but the file has a different size, version, or language, ACE23 fails.

The ACE23 Error String is displayed in the following format:

The file [EXECUTABLE_FILENAME, SIZE, VERSION, LANGUAGE] installed to location [PATHNAME] by component [COMPONENT1] in Package [PACKAGE1] conflicts with the same file in component [COMPONENT2] in Package [PACKAGE2].



Note • ACE23 can be used in conflict analysis when MSI Package, OS Snapshot File, or Marimba NCP File data types are selected as either source or target packages. However, if you specify only Marimba NCP or OS Snapshot files as your source packages, ACE23 and ACE24 will be the only ACEs that you can select on the [Summary Panel](#) of the Conflict Wizard. All other ACEs will be unavailable.

Summary

Table 13-35 • ACE23 Summary

Topic	Description
Conflict Type:	Conflict Rule / Files
Description:	Checks to see if files with the same name and destination directory have the same size, version, and language.
Result:	If a file with the same name and destination directory is found in both the source and target package (and, in the case of an MSI package comparison, the packages have different ComponentId values), but the file has a different size, version, or language, ACE23 fails. If the Source and Target packages are MSIs and have the same Component Id value, no error is reported.
Resolution Type:	Manual
Resolution:	Investigate and decide which file has precedence. If the source file has precedence over the target file, remove the target file.

Cause

A file with the same name and destination directory is found in both the source and target package, but the file has a different size, version, or language.

Resolution

This conflict requires a manual resolution. Use ONE of the following solutions:



Task: *Replace the file in the source package with the same file found in the target package*

1. Retrieve a copy of the file from the operating system in which the target package was taken.
2. Open a transform file or MSI package in InstallShield Editor.
3. Once the project is open, navigate to the **Files and Folders** view, and find the file causing the conflict. Replace that file with the one retrieved from the operating system.
4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE23 again.



Task: *Change the destination of the component that installs the file in the MSI package*

1. Open a transform file or MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Components** view, and select the component that contains the file causing the conflict.

In order to quickly find the component name, go to the File Table via the Direct Editor view and search for the file name. Then, check the component column for the component name.
3. Select the **Destination** field of the component from the **Property** grid and change it to a new destination.

Changing the destination may cause the application to break. Before changing the destination, verify that the application will still work using the new destination.
4. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE23 again.

ACE24

ACE 24 checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value, ACE24 fails.

The ACE24 Error String is displayed in the following format:

The registry entry [REGISTRY_ENTRY] in [PACKAGE1] conflicts with the same registry entry in the [COMPONENT] in [PACKAGE2].



Note • ACE24 can be used in conflict analysis when MSI Package, OS Snapshot File, or Marimba NCP File data types are selected as either source or target packages. However, if you specify only Marimba NCP Files as your source packages, ACE23 and ACE24 will be the only ACEs that you can select on the Conflict Rules Panel of the Conflict Wizard. All other ACEs will be unavailable.

Summary

Table 13-36 • ACE24 Summary

Topic	Description
Conflict Type:	Conflict Rule / Registry
Description:	Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value.
Result:	If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target packages, but the registry entry has a different data type or value, ACE24 fails.
Resolution Type:	Manual
Resolution:	Investigate and decide which registry entry has precedence. If the source registry entry has precedence over the target registry entry, remove the target registry entry.

Cause

A registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task:

To resolve this conflict:

1. Open a transform file or MSI package in InstallShield Editor.
2. Once the project is open, navigate to the **Registry** view, and find the registry value causing the conflict. Replace that value with the same registry value and data type found in the target package.
3. Select **Save As** from the **File** menu and save the changes as a **Windows Installer Transform (.mst)** file.
4. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the **Conflict Wizard** to check it against ACE24 again.

ACE25

ACE25 checks the entries in the CustomAction table to identify any hard coded paths. The ACE25 Warning String is displayed in the following format:

The [CUSTOM_ACTION_NAME] custom action has a hard coded directory path of [TARGET_PATH] in its Target field.

Summary

Table 13-37 • ACE25 Summary

Topic	Description
Conflict Type:	Best Practices / Recommended Rules
Description:	Checks the entries in the package's CustomAction table to identify any hard coded paths in the Target field.
Result:	If a package has a hard coded path in the Target field of the CustomAction table, a warning is generated.
Resolution Type:	Manual
Resolution:	Open the package file in InstallShield Editor and change the hard coded paths to relative paths in the CustomAction table.

Resolution

Open the package file in InstallShield Editor, and in the Target field of the CustomAction table, change any hard coded paths to relative paths.

ACE26

ACE26 checks the Merge Modules in a package to confirm that they also exist in the Application Catalog. The ACE26 Warning String is displayed in the following format:

The [MERGE_MODULE_NAME], version [VERSION] Merge Module is included with this package and yet not imported into the Application Catalog. It is recommended that all Merge Modules be imported into the Application Catalog.



Note • ACE26 and ACE36, optional *Best Practice ACEs*, both check for conflicts with Merge Modules. ACE26 checks merge modules listed in the **ModuleSignature** table while ACE36 checks the **ModuleDependency** table.

These ACEs are provided to encourage you to import Merge Modules into the Application Catalog and, by doing so, improve the effectiveness of [ACE12](#), which checks for components that contain files that could be replaced by one of the imported Merge Modules.

Summary

Table 13-38 • ACE26 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Merge Module Integrity
Description:	Checks the Merge Modules in a package to confirm that they have been imported into the Application Catalog.
Result:	If a package refers to a Merge Module that does not exist in the Application Catalog, a warning is generated.
Resolution Type:	Manual
Resolution:	Import the missing Merge Module into the Application Catalog.

Resolution

Open the package file in Application Manager and import the identified Merge Module into the Application Catalog.

ACE27

The DuplicateFileData ACE checks to see if data exists in the DuplicateFile MSI table that is not being executed with an associated DuplicateFiles action.

The package contains data ([FileKey]) in the 'DuplicateFile' table but not the necessary actions to use this data. You should consider whether a 'DuplicateFiles' or 'RemoveDuplicateFiles' action is needed for your 'InstallExecuteSequence' table.

Summary

Table 13-39 • ACE27 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks to see if data exists in the DuplicateFile MSI table that is not being executed with an associated DuplicateFiles action.
Result:	If a package contains data in the DuplicateFile table but not the necessary actions to use this data, a warning is generated.
Resolution Type:	Manual

Table 13-39 • ACE27 Summary

Topic	Description
Resolution:	Consider whether a DuplicateFiles or RemoveDuplicateFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the DuplicateFile table.

Resolution

Open the MSI file in InstallShield and add a DuplicateFiles or RemoveDuplicateFiles action to the InstallExecuteSequence table, or remove the unused data from the DuplicateFile table.

ACE28

ACE28 checks the entries of the Environment table to identify any hard coded paths. The ACE28 Warning String is displayed in the following format:

The [ENVIRONMENT_TABLE_NAME] Environment table entry has a hard coded directory path of [DIRECTORY_PATH] in its Value field.

Summary

Table 13-40 • ACE28 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks the Value field of the package's Environment table to identify any hard coded paths.
Result:	If any hard coded paths exist in the Value field of a package's Environment table, a warning is generated.
Resolution Type:	Manual
Resolution:	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Environment table to relative paths.

Resolution

Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Environment table to relative paths.

ACE29

ACE29 checks the entries in the IniFile table to identify any hard coded paths. The ACE29 Warning String is displayed in the following format:

The [PATH_NAME]\[INI_FILE_NAME] INI file has a hard coded directory path of [HARD_CODED_PATH] in its Value field.

Summary

Table 13-41 • ACE29 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks the Value field of a package's IniFile table to identify any hard coded paths.
Result:	If the Value field of a package's IniFile table has any hard coded paths, a warning is generated.
Resolution Type:	Manual
Resolution:	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the IniFile table to relative paths.

Resolution

Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the IniFile table to relative paths.

ACE30

The ACE30 is a check for KeyPath conflicts across components. It identifies components with different ComponentIDs installing the same key file to the same directory. The ACE30 Error String is displayed in the following format:

The [SOURCE_COMPONENT_NAME] component in [SOURCE_PRODUCT_NAME] is installing the [SOURCE_PATH_NAME]\[SOURCE_KEYPATH_FILE] file which is also being installed by [TARGET_PRODUCT_NAME] with a different ComponentId.

More precisely, ACE30 checks to determine if different components are installing identically-named key files to the same directory.



Note • The name of the target product is not displayed when viewing this ACE Error String in the Conflicts View.

The ACE executes the following query to check to see if the same key file is being installed with different ComponentIds:

```
Source.ComponentId <> Target.ComponentId AND Source.csFullPath = Target.csFullPath AND  
Source.KeyPath = Target.KeyPath
```

Summary

Table 13-42 • ACE30 Summary

Topic	Description
Conflict Type:	Conflict Rule / Components
Description:	Checks for KeyPath conflicts across components. It identifies components with different ComponentIDs installing the same key file to the same directory.
Result:	If components with different ComponentIDs installing the same key file to the same directory is identified, a warning is generated.
Resolution Type:	Manual
Result:	If a product component is installing a KeyPath file which is also being installed by a product with a different Component Id, the KeyPath ACE fails.

ACE31

The MoveFileData ACE checks to see if data exists in the MoveFile MSI table that is not being executed with an associated MoveFiles action.

```
The package contains data ([FileKey]) in the 'MoveFile' table but not the necessary actions to use  
this data. You should consider whether a 'MoveFiles' action is needed for your  
'InstallExecuteSequence' table.
```

Summary

Table 13-43 • ACE31 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks to see if data exists in the MoveFile MSI table that is not being executed with an associated MoveFiles action.
Result:	If a package contains data in the MoveFile table but not the necessary actions to use this data, a warning is generated.
Resolution Type:	Manual

Table 13-43 • ACE31 Summary

Topic	Description
Resolution:	Consider whether a MoveFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the MoveFile table.

Resolution

Open the MSI file in InstallShield and add a MoveFiles action to the InstallExecuteSequence table, or remove the unused data from the MoveFile table.

ACE32

ACE32 checks the entries of the Registry table to identify any hard coded paths. The ACE32 Warning String is displayed in the following format:

The [REGISTRY_TABLE_ENTRY_NAME] Registry table entry has a hard coded directory path of [DIRECTORY_PATH_NAME] in its Value field.

Summary

Table 13-44 • ACE32 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks entries in the Registry table to identify any hard coded paths in its Value field.
Result:	If there are any hard coded paths in the Value field of the package's Registry table, a warning is generated.
Resolution Type:	Manual
Resolution:	Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Registry table to relative paths.

Resolution

Open the package file in InstallShield Editor and change any hard coded paths in the Value field of the Registry table to relative paths.

ACE33

The RemoveFileData ACE checks to see if data exists in the RemoveFile MSI table that is not being executed with an associated RemoveFiles action.

The package contains data ([FileKey]) in the 'RemoveFile' table but not the necessary actions to use this data. You should consider whether a 'RemoveFiles' action is needed for your 'InstallExecuteSequence' table.

Summary

Table 13-45 • ACE33 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks to see if data exists in the RemoveFile MSI table that is not being executed with an associated RemoveFiles action.
Result:	If a package contains data in the RemoveFile table but not the necessary actions to use this data, a warning is generated.
Resolution Type:	Manual
Resolution:	Consider whether a RemoveFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveFile table.

Resolution

Open the MSI file in InstallShield and add a RemoveFiles action to the InstallExecuteSequence table, or remove the unused data from the RemoveFile table.

ACE34

The RemoveIniFileData ACE checks to see if data exists in the RemoveIniFile MSI table that is not being executed with an associated RemoveIniFiles action.

The package contains data ([RemoveIniFile]) in the 'RemoveIniFile' table but not the necessary actions to use this data. You should consider whether a 'RemoveIniFiles' action is needed for your 'InstallExecuteSequence' table.

Summary

Table 13-46 • ACE34 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks to see if data exists in the RemoveIniFile MSI table that is not being executed with an associated RemoveIniFiles action.
Result:	If a package contains data in the RemoveIniFile table but not the necessary actions to use this data, a warning is generated.
Resolution Type:	Manual
Resolution:	Consider whether a RemoveIniFiles action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveIniFile table.

Resolution

Open the MSI file in InstallShield and add a RemoveIniFiles action to the InstallExecuteSequence table, or remove the unused data from the RemoveIniFile table.

ACE35

The RemoveRegistryData ACE checks to see if data exists in the RemoveRegistry MSI table that is not being executed with an associated RemoveRegistryValues action.

The package contains data ([RemoveRegistry]) in the 'RemoveRegistry' table but not the necessary actions to use this data. You should consider whether a 'RemoveRegistryValues' action is needed for your 'InstallExecuteSequence' table.

Summary

Table 13-47 • ACE35 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Recommended Rules
Description:	Checks to see if data exists in the RemoveRegistry MSI table that is not being executed with an associated RemoveRegistryValues action.
Result:	If a package contains data in the RemoveRegistry table but not the necessary actions to use this data, a warning is generated.
Resolution Type:	Manual

Table 13-47 • ACE35 Summary

Topic	Description
Resolution:	Consider whether a RemoveRegistryValues action is needed for your InstallExecuteSequence table. If so, open the MSI file in InstallShield and add the appropriate action to the InstallExecuteSequence table, or remove the unused data from the RemoveRegistry table.

Resolution

Open the MSI file in InstallShield and add a RemoveRegistryValues action to the InstallExecuteSequence table, or remove the unused data from the RemoveRegistry table.

ACE36

ACE36 checks the required Merge Modules in a package to confirm that they also exist in the Application Catalog. The ACE36 Warning String is displayed in the following format:

The required [MERGE_MODULE_NAME], version [REQUIRED_VERSION] Merge Module is included with this package and yet not imported into the Application Catalog. It is recommended that all Merge Modules be imported into the Application Catalog.



Note • ACE26 and ACE36, optional *Best Practice ACEs*, both check for conflicts with Merge Modules. ACE26 checks merge modules listed in the **ModuleSignature** table while ACE36 checks the **ModuleDependency** table.

These ACEs are provided to encourage you to import Merge Modules into the Application Catalog and, by doing so, improve the effectiveness of [ACE12](#), which checks for components that contain files that could be replaced by one of the imported Merge Modules.

Summary

Table 13-48 • ACE36 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Merge Module Integrity
Description:	Checks to see if a package's required Merge Modules have been imported into the Application Catalog.
Result:	If any of a package's required Merge Modules do not exist in the Application Catalog, a warning is generated.
Resolution Type:	Manual
Resolution:	Import the package's missing Merge Modules into the Application Catalog.

Resolution

Import the package's missing Merge Modules into the Application Catalog.

WTS01

This ACE examines a Windows Installer package for compatibility with Windows Terminal Server. Windows Terminal Server generally requires installation components to be installed for all users, rather than on a per-user basis.



Note • If your package is not targeting the Windows Terminal Server environment, then you do not need to run this ACE.

WTS01 checks to see if the ALLUSERS property is defined. The WTS01 warning string is displayed in the following format:

The product [PRODUCT_NAME] ('[VERSION]') is currently configured to be installed as per-user. This value affects deployment of this package in a terminal server environment.

Summary

Table 13-49 • WTS01 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Windows Terminal Server Compatibility
Description:	WTS01 checks to see if the ALLUSERS property is defined.
Result:	If the ALLUSERS property is defined, a warning is generated.
Resolution Type:	Automatic (WTSFIX01)
Resolution:	The ALLUSERS property is set to 1.

WTS02

This ACE examines a Windows Installer package for compatibility with Windows Terminal Server. Windows Terminal Server generally requires installation components to be installed for all users, rather than on a per-user basis.



Note • If your package is not targeting the Windows Terminal Server environment, then you do not need to run this ACE.

WTS02 checks for any registry entries installed to per-user specific locations. The WTS02 warning string is displayed in the following format:

The component [COMPONENT_NAME] in package [PACKAGE_NAME]('[VERSION']) contains a per-user registry key [KEY_NAME] with KeyPath [KEYPATH_VALUE]. This value affects deployment of this package in a terminal server environment.

Summary

Table 13-50 • WTS02 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Windows Terminal Server Compatibility
Description:	WTS02 checks for any registry entries installed to per-user specific locations.
Result:	If a registry entry is installed to a per-user specific location, a warning is generated.
Resolution Type:	Automatic (WTSFIX02)
Resolution:	Clears the KEYPATH entry for these resources to ensure that Windows Installer repair mode will not be invoked.

WTS03

This ACE examines a Windows Installer package for compatibility with Windows Terminal Server. Windows Terminal Server generally requires installation components to be installed for all users, rather than on a per-user basis.



Note • If your package is not targeting the Windows Terminal Server environment, then you do not need to run this ACE.

WTS03 checks for any file entries installed to per-user specific locations. The WTS03 warning string is displayed in the following format:

The component '[COMPONENT_NAME]' in package '[PACKAGE_NAME]('[VERSION'])' has a per-user destination '[PATH_NAME]'. This value affects deployment of this package in a terminal server environment.

Summary

Table 13-51 • WTS03 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Windows Terminal Server Compatibility
Description:	WTS03 checks for any file entries installed to per-user specific locations.

Table 13-51 • WTS03 Summary

Topic	Description
Result:	If any file entries are installed to per-user specific locations, a warning is generated.
Resolution Type:	Automatic (WTSFIX03)
Resolution:	Clears the KEYPATH entry for these resources to ensure that Windows Installer repair mode will not be invoked.

WTS04

This ACE examines a Windows Installer package for compatibility with Windows Terminal Server. Windows Terminal Server generally requires installation components to be installed for all users, rather than on a per-user basis.



Note • If your package is not targeting the Windows Terminal Server environment, then you do not need to run this ACE.

WTS04 checks for any ODBC data source entries installed to per-user specific locations. The WTS04 warning string is displayed in the following format:

The component '[COMPONENT_NAME]' in package '[PACKAGE_NAME]'('[VERSION]') has a per-user ODBC Data Source '[NAME]'('[VALUE]')'. This value affects deployment of this package in a terminal server environment.

Summary

Table 13-52 • WTS01 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Windows Terminal Server Compatibility
Description:	WTS04 checks for any ODBC data source entries installed to per-user specific locations.
Result:	If an ODBC data source entry is installed to a per-user specific location, a warning is generated.
Resolution Type:	Automatic (WTSFIX04)
Resolution:	Clears the KEYPATH entry for these resources to ensure that Windows Installer repair mode will not be invoked.

WTS05

This ACE examines a Windows Installer package for compatibility with Windows Terminal Server. Windows Terminal Server generally requires installation components to be installed for all users, rather than on a per-user basis.



Note • If your package is not targeting the Windows Terminal Server environment, then you do not need to run this ACE.

WTS05 checks for any environment strings installed to per-user specific locations. The WTS05 warning string is displayed in the following format:

The component '[COMPONENT]' in package '%s'('%s') has a per-user Environment Setting '[NAME]'(''[VALUE]''). This value affects deployment of this package in a terminal server environment.

Summary

Table 13-53 • WTS05 Summary

Topic	Description
Conflict Type:	Best Practice Rule / Windows Terminal Server Compatibility
Description:	WTS05 checks for any environment strings installed to per-user specific locations.
Result:	If an environment string is installed to a per-user specific locations, a warning is generated.
Resolution Type:	Manual
Resolution:	Duplicate the per-user Environment table entries and conditionalize the installation of these new Environment table entries to only be installed if the ALLUSERS property is set to 1.

User-Defined ACEs

In addition to business rules (ACEs) shipped with ConflictSolver, you can also create your own company-specific business rules to use when detecting conflicts. For example, your organization may want to identify (and change) any VBScript custom actions that have a hard-coded drive letter, any applications that create desktop or uninstall shortcuts, any applications that have Startup registry entries, or any applications that place files in the system directory. You can create custom rules to identify these (and many more company-specific situations) using ConflictSolver.

Creation of user-defined ACEs is done using the [Rules Wizard](#).

Types of User-Defined ACEs

In addition to the ACEs included with ConflictSolver, you can also create your own user-defined ACEs to use when detecting conflicts. You can create three types of ACEs:

Custom - Source Only Packages ACEs

Custom - Source Only Packages ACEs allow you to quickly test any column or any value of a table to support your business logic. For example, you could use a user-defined ACE to identify packages that create a desktop icon. To define a Source Only Packages ACE, you must define an SQL “Where” clause.

For an example of this type of user-defined ACE, see [Creating a Custom/Source Only Packages ACE](#).



Note • ConflictSolver supports external package conflict checking for Custom - Source Only Packages ACEs. The Source package can be selected from the Application Catalog Database or from an external MSI package.

Custom - Source and Target Packages ACEs

Custom - Source and Target Packages ACEs allow you to compare columns or values of Source package tables (new packages that you want to install onto a user's system) to columns or values of Target package tables (packages already installed on a user's system).

For example, you could use a Source and Target Packages ACE to determine if the installation of a Source package onto a Target system would overwrite or conflict with an existing entry in the .ini file in the System directory of the Target system.

To define a Source and Target Packages ACE, you must define an SQL “Where” clause, and specify a Join Column—a table column in the Application Catalog database that has a matching value for both the Source and Target packages. Rows in each of the packages that have a matching value in the Join Column are selected and those rows are checked against the Source and Target Packages.

For an example of this type of user-defined ACE, see [Creating a Custom/Source and Target Packages ACE](#).



Note • ConflictSolver does not support external package conflict checking for Custom - Source and Target Packages ACEs. Both the Source and Target Packages must be selected from the Application Catalog Database.

DLL - User Provided DLL Based ACEs

DLL - User Provided DLL Based ACEs allow you to run more complex tests—testing many tables in any combination. For example, you could use a DLL-Based ACE to confirm that a source product language is the same as all target product languages. To define a DLL-Based ACE, you use SQL and various programming languages to construct a Windows DLL. With DLL-Based ACEs, you can use a Conflict Application Resolution Definition (CARD) to fix the conflict.

For an example of this type of user-defined ACE, see [Creating a User Provided DLL-Based ACE](#).

Where You Create User-Defined ACEs

When creating ACEs, you need to provide basic information for display in the Rules viewer, Conflicts tab of the Options dialog box, and in the Output Window. You must associate a table with the ACE. You also must categorize the ACE (either using existing ACE categories or by creating your own).

Creation of User-Defined ACEs is done using the [Rules Wizard](#). You can view and edit ACEs by launching the [Rules Viewer Dialog Box](#) and opening an ACE in the [ACE Rule Properties Dialog Box](#).

Creating User-Defined ACEs

The section describes how to create the three types of user-defined ACEs:

- [Creating a Custom/Source Only Packages ACE](#)
- [Creating a Custom/Source and Target Packages ACE](#)
- [Creating a User Provided DLL-Based ACE](#)

Creating a Custom/Source Only Packages ACE

You can create a user-defined ACE to apply to Source Only Packages. One common task you may want to create a Source Only Packages ACE to handle is to identify packages which create a desktop icon.

ConflictSolver supports external package conflict checking for Custom - Source Only Packages ACEs. The Source package can be selected from the Application Catalog database or from an external MSI package.



Task: *To create a Source Only Packages ACE that identifies desktop icon creation:*

1. Launch ConflictSolver.
2. From the **Conflicts** menu, select **Rules Viewer**. The **Rules Viewer** dialog box opens.
3. Click **New** to launch the **Rules Wizard**. The **Welcome Panel** opens.
4. Click **Next**. The **General Information Panel** opens.
5. Input the following values:

Option	Value
Name	MYACE
Associated Table	csmsiShortcut
Brief Description	MYACE - Find desktop icons
Description	Locates package that create desktop icons.
Information URL	http://www.yourcompany.com/MYACE.htm

6. Click **Next**. The **Additional Information Panel** opens.

7. Input the following values:

Option	Value
Category	Shortcuts
Rule Type	Custom - Source Only Packages

8. Click **Next**. The **Custom Options Panel** opens.

9. Input the following values:

Option	Value
Error String	Desktop icons [Name] are not allowed.
Display String	A desktop icon called [Name] is created.
Severity	Warning
Report 'No' Results	Deselected

In the example above, [Name] is a token. Tokens allow you to insert values at run-time from the internal Application Catalog Database or from an external MSI package into the Error or Display string. To use token replacement in a string, click the arrow to the right of the Error String and Display String text boxes and select a column name from the list. The column name is then inserted into the string in the following format: [ColumnName].

The Token list is provided for your convenience; if you prefer, you can type the tokens directly in the text boxes. You could also use the [ProductName] pseudo-token to insert the name of the package in a message, even though ProductName is not a table column name.



Note • For more information, see [Token Grammar](#)

10. Click **Next**. The **Where Clause Panel** opens.

11. In the **Where Clause Panel**, click **Build Expression**. The **Expression Builder** dialog box opens.

12. In the **Expression Builder**, input the following values:

Option	Value
Table Columns	[Directory_]
Comparison Operator	= (Equal To)
String Constant	DesktopFolder

13. Click OK to close the **Expression Builder** and return to the **Where Clause Panel**. The expression that you just built is now displayed in the Where Clause text box:

```
[Directory_] = 'DesktopFolder'
```

When you are constructing simple expressions, it is helpful to use the Expression Builder dialog box, but you are not limited to the formatting options that the Expression Builder provides to you. If you know how to write Where clauses in SQL, you can use significantly more powerful expressions by entering them directly in the Where Clause text box on the Where Clause Panel or on the Where Clause Tab of the [ACE Rule Properties Dialog Box](#).

14. Click **Test** to validate the expression.
15. Click **Next**. The **Summary Panel** opens.
16. On the **Summary Panel**, review the summary of your new ACE and click **Finish**.
17. Click **Close** to exit the **Rules Viewer**. This new user-created ACE is now available for use in subsequent conflict identification.

Creating a Custom/Source and Target Packages ACE

You can create a custom ACE to apply to Source and Target Packages. For example, you could use a Source and Target Packages ACE to determine if the installation of a Source package onto a Target system would overwrite or conflict with an existing entry in the .ini file in the System directory of the Target system.



Caution • ConflictSolver does not support external package conflict checking for Custom - Source and Target Packages ACEs. Both the Source and Target Packages must be selected from the ConflictSolver Application Catalog Database. If you attempt to run a conflict check on an external MSI package using a Source and Target Packages ACE, that custom ACE will not be executed.



Task: *To create a Source and Target Packages ACE that identifies .ini file conflicts:*

1. Launch ConflictSolver.
2. From the **Conflicts** menu, select **Rules Viewer**. The **Rules Viewer** dialog box opens.
3. Click New to launch the **Rules Wizard**. The **Welcome Panel** opens.
4. Click **Next**. The **General Information Panel** opens.
5. Input the following values:

Option	Value
Name	INICheck
Associated Table	csmsilniFile

Option	Value
Brief Description	INICheck - Identifies conflicts found in an .ini file.
Description	Determines if the installation of a Source package onto a Target system would overwrite or conflict with an existing entry in the .ini file in the Target system's System directory.
Information URL	http://www.yourcompany.com/INICheck.htm

6. Click **Next**. The **Additional Information Panel** opens.
7. Input the following values:

Option	Value
Category	INI Files
Rule Type	Custom - Source and Target Package

8. Click **Next**. The **Custom Options Panel** opens.
9. Input the following values:

Option	Value
Error String	The INI file called [Source.FileName] in the directory [Source.csFullPath] writes to the [Source.Section] section which is also written by the target package, [Target.ProductName].
Display String	The INI file called [Source.FileName] in the directory [Source.csFullPath] writes to the [Source.Section] section.
Severity	Warning
Report 'No' Results	Deselected

In the example above, [Source.FileName], [Source.csFullPath] and [Source.Section] are tokens. Tokens allow you to insert values at runtime from the internal Conflict Solver Application Catalog Database into the Error or Display string. To use token replacement in a string, click the arrow to the right of the Error String or Display String text boxes and select a column name from the list. The column name is then inserted into the string in the format of [Source.ColumnName] or [Target.ColumnName], with the prefix identifying whether the column is in the Source or Target package.



Note • If no prefix is used, ConflictSolver assumes the "Source." prefix.

You can also use the [Target.ProductName] and [Source.ProductName] pseudo-tokens to insert the name of the Source or Target package in a message, even though ProductName is not a table column name.



Note • The Token list is provided for your convenience; if you prefer, you can type the variables directly in the text boxes. For more information, see [Token Grammar](#).

10. Click **Next**. The **Where Clause Panel** opens.
11. Click **Build Expression**. The **Expression Builder** dialog box opens.
12. Input the following values:

Option	Value
Table Columns	[Source].[csFullPath]
Comparison Operator	= (Equal To)
String Constant	SystemFolder

13. Click **OK** to close the **Expression Builder** and return to the **Where Clause Panel**. The expression that you just built is now displayed in the Where Clause text box:

[Source].[csFullPath] = 'SystemFolder'



Note • When you are constructing simple expressions, it is helpful to use the Expression Builder dialog box, but you are not limited to the formatting options that the Expression Builder provides to you. If you know how to write Where clauses in SQL, you can use significantly more powerful expressions by entering them directly in the **Where Clause** text box on the **Where Clause Panel** or on the **Where Clause Tab** of the [ACE Rule Properties Dialog Box](#).

14. Click Build Expression again to open the **Expression Builder**.
15. Input the following values:

Option	Value
Table Columns	[Source].[Section]
Comparison Operator	= (Equal To)
String Constant	[Target].[Section]
Expression Operator	AND



Note • When using the Expression Builder dialog box to create a Source and Target Packages custom ACE to compare the value of a column in the source table to the value of a column in the target table, you can select the first table column name from the Table Columns list. However, you have to manually enter the second table

column name in the Constant text box. When doing so, enter the table column name using the same syntax that is used in the Table Columns list: [Source].[ColumnName] or [Target].[ColumnName].

16. Click **OK** to close the **Expression Builder** and return to the **Where Clause Panel**. The expression that you just built is now displayed in the **Where Clause** text box, added to the end of the first expression you built:

```
[Source].[csFullPath] = 'SystemFolder' AND  
[Source].[Section] = '[Target].[Section]'
```

17. Select **csFullPath** from the **Join Column** list.

The **Join Column** is a table column in the Application Catalog database that has a matching value for both the Source and Target packages. Rows in each of the packages that have a matching value in the Join Column are selected and those rows are checked against the Source and Target Packages.

18. Click **Test** to validate the expression. A message appears stating that the query executed properly.
19. Click **Next**. The **Summary Panel** opens.
20. Review the summary of your new ACE and click **Finish**.
21. Click **Close** to exit the **Rules Viewer**. This new Custom ACE is now available for use in subsequent conflict identification.

Creating a User Provided DLL-Based ACE

To demonstrate how to create a DLL-Based ACE, an ACE SDK was installed with AdminStudio, in the following directory:

AdminStudio Installation Directory\ConflictSolver\ACESDK

Included in this directory is a VC++ 6.0 DLL-based project which is a fully formed example ACE. This example includes utility functions that allow you to integrate your own ACE execution within our conflict persistence model.

Specifying the Visual Studio C++ Type Library File Path

ConflictSolver provides a data structure to each ACE constructed using the ACE SDK. This data structure includes an ADO Connection interface which is the means by which you can execute queries against the Application Catalog.

In order to build the ACE SDK, you will need to make sure that Visual Studio can locate the needed files for ADO.

The *first time* you create a DLL-based ACE, you need to open Visual Studio C++ and specify the path for the type library file (msado15.dll) by performing the following steps:



Task: *To specify the Visual Studio C++ type library file path:*

1. Launch Visual Studio C++ 6.0.
2. Select **Options** from the **Tools** menu. The **Options** dialog box opens.
3. On the **Directories** tab, set the **Platform** field to **Win32** and the **Show directories for** field to **Library files**.

- Under **Directories**, specify the location of the needed files for an ADO Connection interface. The directory specified below is a likely common directory for storing these files:

C:\Program Files\Common Files\System\ADO

- Click **OK** to exit the **Options** dialog box.



Caution • If you fail to specify the correct Library files path, you will encounter the following error when building an ACESDK project with Visual Studio C++:

fatal error C1083: Cannot open type library file: 'msado15.dll'; No such file or directory

Creating a DLL-Based ACE

To learn how to create a DLL-Based ACE, use the ACE SDK files to perform the following steps:



Task: To create a DLL-Based ACE:

- Launch Windows Explorer and navigate to the following directory:
AdminStudio Installation Directory\ConflictSolver\ACESDK
- Copy this folder and its contents and store it in a convenient location.
- Launch VC++ 6.0 and open the ACESDK.dsp project file within that newly created folder.
- Review the code and make any desired changes.
- Still in VC++ 6.0, build the ACESDK.dsp project to create a new .DLL file.
- Launch ConflictSolver.
- From the **Conflicts** menu, select **Rules Viewer**. The **Rules Viewer** dialog box opens.
- Click **New** to launch the **Rules Wizard**. The Rules Wizard **Welcome Panel** opens.
- Click **Next**. The **General Information Panel** opens.
- Input the following values:

Option	Value
Name	ACELanguage
Associated Table	csmsiProperty (This is the ConflictSolver table associated with this example ACE.)
Brief Description	ACELanguage - Check product language consistency.
Description	Confirm that source product language is the same as all target product languages.

Option	Value
Information URL	http://www.yourcompany.com/ACELanguage.htm

11. Click **Next**. The **Additional Information Panel** opens.
12. Input the following values:

Option	Value
Category	Type in a new category name: Product Language
Rule Type	DLL - User Provided DLL

13. Click **Next**. The **DLL-Based ACEs Panel** opens.
14. Input the following values:

Option	Value
ACE/CARD DLL File	Click Browse and select the DLL that you built in Step 5 above.
ACE Function Name	ExampleACE (as designed in the sample)
CARD Function Name	ExampleCARD (as designed in the sample)

15. Click **Test** next to **ACE Function Name** or **CARD Function Name** to validate that the exported function does exist.
16. Click **Next**. The **Summary Panel** opens.
17. On the **Summary Panel**, review the summary of your new ACE and click **Finish**.
18. Click **Close** to exit the **Rules Viewer**. This new DLL-Based ACE is now available for use in subsequent conflict identification.

Editing User-Defined ACEs



Task: *To edit a user-defined ACE:*

1. From the **Conflicts** menu, select **Rules Viewer**. The **Rules Viewer** dialog box opens.
2. Select the user-defined ACE you want to modify and click **Edit**. The **ACE Rule Properties** dialog box opens.
3. In the **ACE Rule Properties** dialog box, edit the options as necessary.
4. Click **OK** to close the dialog box.
5. Click **Close** to exit the **Rules Viewer** dialog box.

The modified ACE is available for use in subsequent conflict identification.

Deleting User-Defined ACEs



Task: *To delete a user-defined ACE:*

1. From the **Conflicts** menu, select **Rules Viewer**. The **Rules Viewer** dialog box opens.
2. Select the ACE you want to remove and click **Delete**.
3. Confirm the deletion.

The user-defined ACE is removed from the available rules. If this ACE was the only one in a user-defined ACE category, the category will be removed when you close the Rules Viewer dialog box.

Viewing ACE Metrics

When ACEs are run, ConflictSolver generates metrics and logs them in the AceLog.txt file, located in the following directory:

AdminStudio Shared\ConflictSolver\AceLog.txt

The following is an example of the beginning of an AceLog.txt file:

11/06/03 09:26:35 ACE/CARD Execution started

ACE03

```
SELECT DISTINCT csmsiComponent.Component,  
                csmsiComponent.ComponentId,csmsiComponent.RowID  
FROM csmsiComponent WHERE csmsiComponent.PkgRowID_ = 5
```

Records returned =>(124)
Query Time =>(0.23 seconds)

```
SELECT csmsiFile.FileName FROM csmsiFile WHERE  
       csmsiFile.Component_ = 'DeletedLinks.10' AND  
       csmsiFile.PkgRowID_ = 5
```

ORDER BY csmsiFile.FileName

Records returned =>(0)

Query Time =>(0.07 seconds)

Time taken to execute (ACE03) : 1.06 seconds

ACE07

```
SELECT [fs].[RowID], [ft].[RowID], [ft].[PkgRowID_],
       [ft].[FileName], [cs].[Component], [ct].[Component],
       [cs].[ComponentId], [ct].[ComponentId], [fs].[Version],
       [fs].[FileSize], [fs].[Language] FROM (([csmsiFile] AS [fs]
INNER JOIN [csmsiFile] AS [ft] ON ( [fs].[FileName] =
[ft].[FileName] AND [fs].[Version] = [ft].[Version] AND
[fs].[FileSize] = [ft].[FileSize] AND [fs].[Language] =
[ft].[Language])) ) INNER JOIN [csmsiComponent] AS [cs] ON
[fs].[PkgRowID_] = [cs].[PkgRowID_] AND [cs].[Component] =
[fs].[Component_]) INNER JOIN [csmsiComponent] AS [ct] ON
[ft].[PkgRowID_] = [ct].[PkgRowID_] AND [cs].[ComponentId] <>
[ct].[ComponentId] AND [cs].[csFullPath] = [ct].[csFullPath] AND
[ct].[Component] = [ft].[Component_] WHERE [fs].[PkgRowID_] = 5
AND [ft].[PkgRowID_] IN (4)
```

Records returned =>(0)

Query Time =>(0.10 seconds)

Time taken to execute (ACE07) : 0.74 seconds

ACE08

```
SELECT [cs].[Component] AS [SrcComponent], [fs].[RowID] AS [SrcRowID],
       [ft].[RowID] AS [TargetRowID], [ft].[PkgRowID_] AS
       [TargetPkgRowID], [fs].[Version] AS [SrcVersion],
       [fs].[FileName], [cs].[csFullPath], [ct].[Component] AS
       [TargetComponent], [ft].[Version] AS [TargetVersion] FROM
       (([csmsiFile] AS [fs] INNER JOIN [csmsiFile] AS [ft] ON
[fs].[FileName] = [ft].[FileName]) INNER JOIN [csmsiComponent]
AS [cs] ON [cs].[PkgRowID_] = [fs].[PkgRowID_] AND
[cs].[Component] = [fs].[Component_]) INNER JOIN
[csmsiComponent] AS [ct] ON [cs].[ComponentId] =
[ct].[ComponentId] AND [ct].[PkgRowID_] =
[ft].[PkgRowID_] AND [ct].[Component] = [ft].[Component_] AND
[cs].[csFullPath] = [ct].[csFullPath] WHERE
[fs].[PkgRowID_] = 5 AND [ft].[PkgRowID_] IN (4) AND
([fs].[Version] <> [ft].[Version] OR ([fs].[Version] IS
NULL AND [ft].[Version] IS NOT NULL) OR ([ft].[Version]
IS NULL AND [fs].[Version] IS NOT NULL))
```

Records returned =>(0)

Query Time =>(0.09 seconds)

Time taken to execute (ACE08) : 0.73 seconds

Location of ACE Files

ACE information is stored in three files that are installed with ConflictSolver.

Table 13-54 • ACE File Names and Locations

Type	File Name	Installation Location
Standard ACEs	isconflict.ace	AdminStudio installation directory in the following folder: ConflictSolver/Support
Merge Module ACEs	ismmconflict.ace	AdminStudio installation directory in the following folder: ConflictSolver/Support
Custom ACE File	CustomConflictFile.ace	AdminStudio Shared directory

ConflictSolver requires that the Standard ACE and Merge Module ACE files must remain in their installed location. However, the location of the Custom ACE File can be changed by opening the **Rules** tab of the **Options** dialog box, and editing the **Custom ACE Rule File** field.

Conflict Application Resolution Definitions (CARDS)

Conflict Application Resolution Definitions (CARDS) are the rules used to fix conflicts identified by [Application Conflict Evaluators \(ACEs\)](#) when conflict detection is performed. Before a CARD is run, you are given the opportunity to review how the CARD will resolve the conflict and decide whether or not to perform the resolution. See [Conflict Resolution Process](#) for more information.

The following links, grouped by conflict type, explain each CARD.

Table 13-55 • CARD Categories

Conflict Type	Associated Resolutions
Component	CARD02, CARD04, CARD05, CARD06
Files	CARD07
ODBC Resources	CARD15
Product Properties	CARD18, CARD19, CARD20

CARD Index

The following list provides links to each individual Conflict Application Resolution Definition (CARD) and its associated ACE (Application Conflict Evaluators). These CARDs are used to resolve conflicts between installation packages that were identified by the CARD's corresponding ACE.

Conflicts found by ACEs that do not have associated CARDs must be resolved manually. See [ACE Index](#) for more information.

Table 13-56 • CARD Index

CARD	Brief Description of ACE	Action Taken by CARD
CARD02 for ACE02	Checks to see if components in different packages that have matching ComponentIds also have identical destination paths.	The destination path of the component in the Source package is automatically set to match that of the component in the Target package.
CARD04 for ACE04	Checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the CreateFolder table.	A CreateFolder entry is created for the component.
CARD05 for ACE05	Checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package.	Modifies the component so that only one EXE or DLL exists, and it adds new components for remaining EXE, DLL, OCX, HLP, CHM, TLB, SYS, and DRV files.
CARD06 for ACE06	Checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file.	The executable module is automatically made the key file.
CARD07 for ACE07	Checks for the existence of the same file in components with different ComponentIds.	The Source package ComponentId is changed to match the Target package Component Id.
CARD15 for ACE15	Checks for the existence of identical ODBC entries in components with different ComponentIds.	ConflictSolver changes the Source ComponentId to match that of the Target ComponentId.
CARD18 for ACE18	Checks the Package Code to see if it is unique.	A new, unique Package Code is automatically generated to replace the duplicate code.

Table 13-56 • CARD Index

CARD	Brief Description of ACE	Action Taken by CARD
CARD19 for ACE19	Checks the Product Code to see if it is unique.	A new, unique Product Code is automatically generated to replace the duplicate code.
CARD20 for ACE20	Checks the Upgrade Code to see if it is unique.	A new, unique Upgrade Code is automatically generated to replace the duplicate code.



Note • Before a CARD is run, you are given the opportunity to review how the CARD will resolve the conflict and decide whether or not to perform the resolution. See [Conflict Resolution Process](#) for more information

CARD02

Conflict

CARD02 resolves [ACE02](#), which checks to see if components in different packages that have matching ComponentIds also have identical destination paths. If components with the same ComponentId have different destination paths, ACE02 fails.

Resolution

CARD02 automatically sets the destination path of the component in the Source package to match that of the component in the Target package. To do this, CARD02 runs the following query and replaces the Directory_ column with a run-time translation of the targets component (csFullPath) path:

```
MsidBUtils::GetDirectoryTargetPathKey.  
SELECT `Directory_` FROM `Component`  
WHERE ComponentId='Source ComponentId'
```

CARD04

Conflict

CARD04 resolves [ACE04](#), which checks to see if components that have NULL KeyPath values are valid by checking for an entry for the component in the [CreateFolder](#) table. If a component does not have a key file, and if the directory/component pair is not listed in the CreateFolder table, ACE04 fails.

Resolution

CARD04 creates a CreateFolder entry for the component by executing the following query:

```
INSERT INTO CreateFolder ( `Directory_`, `Component_` )  
VALUES ('Source Directory','Source Component')
```


CARD05

Conflict

CARD05 resolves [ACE05](#), which checks for the existence of more than one executable (EXE, DLL, OCX, HLP, CHM, TLB, SYS, DRV) per component in a Windows Installer package. If more than one executable exists in a component, ACE05 fails.

Resolution

CARD05 modifies the component so that only one EXE or DLL exists, and it adds new components for remaining EXE, DLL, OCX, HLP, CHM, TLB, SYS, and DRV files. To do this, CARD05 generates a new component name and ComponentId and inserts a record in the Component table and FeatureComponents table. The relevant entry in the File table would then be updated to effectively move the file into the new component.

CARD06

Conflict

CARD06 resolves [ACE06](#), which checks whether the executable module (EXE, DLL, OCX, HLP, CHM, TLB, SYS, or DRV) within the component is the key file. If the executable module is not the key file, ACE06 fails.

Resolution

CARD06 automatically makes the executable module the key file. To do this, CARD06 runs the following query and replaces the KeyPath with an entry from the File Table - File Column which is associated with this component and of the correct type.

```
SELECT `KeyPath` FROM `Component` WHERE `
    ComponentId` = 'Source ComponentId'
```

CARD07

Conflict

CARD07 resolves [ACE07](#), which checks for the existence of the same file in components with different ComponentIds. If the same file exists in components with different ComponentIds, ACE07 fails.

Resolution

CARD07 changes the Source package ComponentId to match the Target package Component Id. To do this, CARD07 runs the following query against the source package and then updates the ComponentId with the ComponentId from the target package:

```
SELECT `ComponentId` FROM `Component` WHERE
    Component` = 'Source Package ComponentId'
```

CARD15

Conflict

CARD15 resolves [ACE15](#), which checks for the existence of identical ODBC entries in components with different ComponentIds. If identical ODBC entries exist in components with different ComponentIds, ACE15 fails.

Resolution

CARD15 changes the Source ComponentId to match that of the Target ComponentId. To do this, CARD15 runs the following query against the source package and then updates the ComponentId with the ComponentId from the target package:

```
SELECT `ComponentId` FROM  
Component ` WHERE `Component` = 'Source Package ComponentId'
```

CARD18

Conflict

CARD18 resolves [ACE18](#), which checks the Package Code to see if it is unique. If the Package Code is identical to any other Package Code in ConflictSolver, ACE18 fails.

Resolution

CARD18 updates the PackageCode by inserting a newly generated GUID into the MSI package using MsiSummaryInfo SetProperty.

CARD19

Conflict

CARD19 resolves [ACE19](#), which checks the Product Code to see if it is unique. If the Product Code is identical to any other Product Code in ConflictSolver, ACE19 fails.

Resolution

CARD19 automatically generates a new, unique Product Code to replace the duplicate code. To do this, CARD19 runs the following query.

```
UPDATE Property SET `Value` = 'Newly generated GUID'  
WHERE `Property` = 'ProductCode'
```

CARD20

Conflict

CARD20 resolves [ACE20](#), which checks the Upgrade Code to see if it is unique. If the Upgrade Code is not unique, ACE20 fails.

Resolution

CARD20 generates a new, unique Upgrade Code to replace the duplicate code. To do this, CARD20 runs the following query:

```
UPDATE Property SET `Value` = 'Newly generated GUID'
WHERE `Property` = 'UpgradeCode'
```

ConflictSolver Reference

ConflictSolver Reference includes the same topics displayed when you click a help button from a Wizard or dialog box in the ConflictSolver interface. Reference information for ConflictSolver is organized into the following areas:

Table 13-57 • ConflictSolver Reference Organization

Section	Description
ConflictSolver Interface	This section contains information about the main ConflictSolver interface, including menus and the toolbar.
Views	Each view available in ConflictSolver is covered in this section.
Dialog Boxes	Specific help for each dialog box in ConflictSolver is covered in this section.
Wizards	This section contains a panel-by-panel reference for each Wizard in ConflictSolver.

ConflictSolver Interface

The ConflictSolver user interface consists of three areas: the ConflictSolver Navigation window, the Details pane, and the Output Window. Both the ConflictSolver Navigation window and the Output Window are dockable.

- **Navigation window**—The ConflictSolver Navigation window consists of two tabs: the Product View and the Merge Modules View.
- **Details pane**—When you select different items in the Product View or Merge Modules View, the Details pane displays corresponding information about that item.
- **Output window**—The Output Window consists of tabs where output is displayed during different ConflictSolver processes.

ConflictSolver also supports many context menus in addition to the menu bar and toolbar functionality.

Menus and Toolbar

The following commands and toolbar buttons are available in ConflictSolver. Additionally, several commands can be accessed through [Context Menus](#).

Table 13-58 • Menus & Toolbars

Menu	Command	Shortcut	Button	Description
Catalog	Connect	Ctrl+O		Displays the Connect Application Catalog dialog box, where you can open an existing Application Catalog.
Catalog	Refresh	Ctrl+R		Refreshes the current Product and Merge Module views. This is particularly useful if multiple people are working on the same Application Catalog.
Catalog	Disconnect			Closes the currently open Application Catalog.
Catalog	Remove			
Catalog	Properties			View Application Catalog properties.
Catalog	Log Out			
Catalog	Exit			Closes ConflictSolver.
Edit	Find	Ctrl+F		Use to search for data in various tables in the Application Catalog.
View	Toolbar	Ctrl+0		Toggles the Toolbar.
View	Status Bar	Ctrl+1		Toggles the Status Bar.
View	Output Window	Ctrl+2		Toggles the Output Window.
View	Application Manager	Ctrl+3		Select to switch to the ConflictSolver (Process Assistant).
View	ConflictSolver	Ctrl+4		Select to switch to the ConflictSolver (Process Assistant).
View	Process Assistant	Ctrl+5		Select to switch to the ConflictSolver (Process Assistant).
Conflicts	Validation Wizard	F4		Launches the Validation Wizard, which you can use to select any Windows Installer package for validation. The output is displayed in the Output Window.

Table 13-58 • Menus & Toolbars (cont.)

Menu	Command	Shortcut	Button	Description
Conflicts	Conflict Wizard	F5		Launches the Conflict Wizard, through which you can open a Windows Installer package and start comparing it against packages in ConflictSolver for conflicts. The output is displayed in the Output Window.
Conflicts	Patch Impact Analysis Wizard	F7		Launches the Patch Impact Analysis Wizard, which you can use to obtain the Windows operating system and Microsoft application patches you need and thoroughly test the impact they will have on your environment.
Conflicts	Stop			Stops the validation or conflict search.
Conflicts	Resolve External Package Conflicts	F6		Displays the Conflict Resolution dialog box, through which you resolve conflicts discovered by ConflictSolver between an external package and packages already in ConflictSolver.
Conflicts	Rules Viewer			Displays the Rules Viewer dialog box, from which you can create new ACE rules.
Reports	Package			Creates the Package Report.
Reports	File			Creates the Files Report.
Reports	Registry			Creates the Registry Report.
Reports	Crystal Reports Conflicts			Creates the pre-defined Crystal Reports Conflicts Report.
Reports	Crystal Reports Files			Creates the pre-defined Crystal Reports Files Report.
Reports	Crystal Reports Registry			Creates the pre-defined Crystal Reports Registry Report.
Tools	Customize	Alt + M		Opens the Customize dialog box, where you can customize toolbar settings.

Table 13-58 • Menus & Toolbars (cont.)

Menu	Command	Shortcut	Button	Description
Tools	Options	Alt+T		Opens the Options dialog box, where you can change various ConflictSolver settings including default conflicts to identify, the default location for Windows Installer packages, the default Application Catalog, report locations, the default validation file, and duplicate package identification options.
Help	Contents			Launches the online Help Library and displays the Contents tab.
Help	Index			Launches the online Help Library and displays the Index tab.
Help	Search			Launches the online Help Library and displays the Search tab.
Help	Support Central			Connects to the AdminStudio Support Web site.
Help	Web Community			Connects to the Web Community.
Help	ReadMe			Displays the AdminStudio ReadMe file.
Help	Feedback			Connects to an online form, through which you can provide feedback about ConflictSolver or AdminStudio.
Help	About Conflict Solver			Displays the About dialog box with version information for ConflictSolver.
	OS Snapshot Import			Opens the OS Snapshot Information Panel, which allows you to import an OS snapshot into ConflictSolver. Only available from the toolbar or context (right-click) menu.
	Help Library			Launches the online Help Library.

Product View Icons

The following icons are used on the ConflictSolver and Application Manager Product Views:

Table 13-59 • ConflictSolver and Application Manager Product View Icons




Name	Icon	Description
Application Catalog		The Application Catalog that you are connected to. When you select it, the Application Catalog View opens and displays detailed summary information.
Group		A group, which is used to organize your data.
		A group containing a product with unresolved conflicts. See Resolving Conflicts .

Table 13-59 • ConflictSolver and Application Manager Product View Icons (cont.)

























Name	Icon	Description
Product		A product that has been imported into the Application Catalog database. The product could be an MSI file, an Other Setup Type package, or a Marimba NCP File.
		A product with unresolved conflicts.
		A product that has an unresolved question with its associated files. One or more files, transforms, or patch files associated with a product is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Application Manager Product View identifying the files in question and prompting you to take action to resolve the problem.
		A product that is stored in the Software Repository.
		A product that is stored in the Software Repository and is checked out.
		A product with unresolved conflicts that also has an unresolved question with its associated files. One or more files, transforms, or patch files associated with a product is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Product View describing the missing files and prompting you to locate the missing files. When this icon is selected, a message appears in the Product View identifying the files in question and prompting you to take action to resolve the problem. Also, see Resolving Conflicts .
		A product linked to a Remote Application Catalog. See Automatically Importing Packages .
		A linked product in a local Application Catalog that has been updated on the Remote Server and needs to be refreshed in order to update its data with the data on the Remote Server. See Manually Updating a Linked Package .
Transforms		A transform.
		A transform that has an unresolved question. It is either missing from the original import directory or its last modified date does not match the last modified date stored in the Application Catalog. When this icon is selected, a message appears in the Application Manager Product View prompting you to take action to resolve the problem.

Table 13-59 • ConflictSolver and Application Manager Product View Icons (cont.)

Name	Icon	Description
OS Snapshot		An OS snapshot, which is a file representing a particular computer system's contents.
Conflicts		Click to access the product's Conflicts View, where you can access information pertaining to the last execution of the Conflict Wizard.
Validation		Click to access the product's Validation View, which lists all ICE Errors and Warnings that were generated the last time the selected package was validated.
Patch Impacts		Click to access the product's Patch Impacts Analysis View, which lists patches for which there is patch impact data persisted against the product, and identifies the patch that caused the impact.
Dependencies		Click to display the Dependencies associated with the package. See Dependencies View for more information.
Extended Attributes		Click to display the optional Extended Attributes associated with the package. See Extended Attributes View for more information.
Files or Files/Components		Click to display the Files View or the Files and Components View, listing all of the files and components in the MSI package, merge module, OS snapshot, or NCP file.
		Click to display the Files View or the Files and Components View, listing all of the files and components in the MSI package, merge module, OS snapshot, or NCP file. The question mark icon indicates that a file is missing from the original import directory. When this icon is selected, a message appears in the Files View describing the missing files and prompting you to locate the missing files.
INI File Changes		Click to display the INI File Changes View , listing any INI file changes made by the product.
Merge Modules		Click to display any merge modules included the product.
Associated Patches		Click to display the product's Associated Patches View , which displays a list of imported patches that, if installed, would update the selected product.
Registry		Click to display any registry entries created or changed by the product.
Shortcuts		Click to display any shortcuts created by the product.
Tables		Click to view the data for a given package in the Application Catalog.

Context Menus

ConflictSolver includes several context menus which can be accessed by right-clicking on nodes within the Product View. These menus provide specific functionality in relation to what is clicked.

- [Groups](#)
- [MSI Package](#)
- [Other Setup Types](#)
- [Marimba NCP Files](#)
- [OS Snapshot](#)
- [Patches View on the Patches Tab](#)

Groups

When you right-click on a group in the Product View, the following commands are available through the context menu:

Table 13-60 • ConflictSolver Group Context Menu Commands

Command	Description
Conflict Wizard	Launches the Conflict Wizard directly to the Target Information Panel.
Patch Impact Analysis Wizard	Launches the Patch Impact Analysis Wizard, which you can use to identify conflicts between Microsoft patches and packages and OS Snapshots in your Application Catalog.
Scan for Dependencies	If dependency information has not already been generated by the Patch Impact Analysis Wizard, select this option to generate dependency information for all MSI packages in this group.

MSI Package

When you right-click on an MSI package in the Product View, the following commands are available through the context menu:

Table 13-61 • MSI Package Context Menu Commands

Command	Description
Validate Package	Select to validate the selected package.
Test Package	Select to test the selected package using PackageExpert. See Testing Packages From ConflictSolver for more information.
Conflict Wizard	Launches the Conflict Wizard directly to the Target Information Panel.

Table 13-61 • MSI Package Context Menu Commands

Command	Description
Patch Impact Analysis Wizard	Launches the Patch Impact Analysis Wizard, which you can use to identify conflicts between Microsoft patches and packages and OS Snapshots in your Application Catalog.
Scan for Dependencies	If dependency information has not already been generated by the Patch Impact Analysis Wizard, select this option to generate dependency information for an MSI package.
Associate with Workflow Manager Application	Launches the Associate with Workflow Manager Application dialog box, from which you can pick a package in Workflow Manager with which to associate the extended attribute data for the selected product.
Reports	Select one of the following: <ul style="list-style-type: none"> • Package—Creates the Package Report • File—Creates the Files Report. • Registry—Creates the Registry Report.

Other Setup Types

No context menu is available for an Other Setup Types package (a package containing non-MSI based setup files) in the Product View.

Marimba NCP Files

When you click on an Marimba NCP File package in the Product View, the following commands are available through the context menu:

Table 13-62 • Marimba NCP Files Context Menus

Command	Description
Conflict Wizard	Launches the Conflict Wizard directly to the Target Information Panel.

OS Snapshot

When you right-click on an OS Snapshot in the Product View, the following commands are available through the context menu:

Table 13-63 • OS Snapshot Context Menu Commands

Command	Description
Conflict Wizard	Launches the Conflict Wizard directly to the Target Information Panel.

Table 13-63 • OS Snapshot Context Menu Commands (cont.)

Command	Description
Patch Impact Analysis Wizard	Launches the Patch Impact Analysis Wizard, which you can use to identify conflicts between Microsoft patches and packages and OS Snapshots in your Application Catalog.

Patches View on the Patches Tab

When you click on an individual patch in the Patch View, the following commands are available through the context menu:

Table 13-64 • Patches View Context Menus

Command	Description
Patch Impact Analysis Wizard	Launches the Patch Impact Analysis Wizard, which you can use to identify conflicts between Microsoft patches and packages and OS Snapshots in your Application Catalog.
Generate Report	Select to generate a Patch Report for that patch.

Output Window

When validation is performed and when the Import, Package Auto Import, Patch Impact Analysis, and Conflict Wizards are run, the output messages and results of those wizards are displayed in the various tabs of the Output Window.

Table 13-65 • Output Window Context Menus

Tab	Description
Output Tab	When the Validation Wizard or Conflict Wizard or Patch Impact Analysis Wizard is run, or when validation is performed during import or by using the Validate Package function, messages about the validation or conflict identification processes are displayed in this tab. When the process is complete, the Errors, Warnings, and Info messages (for Validation only) that were generated are listed in a table format in the Validate or Conflicts tab. Identified patch conflicts are listed in table format in the Patch Impact tab.
Import Tab	When the Import Wizard is run to import a package into the Application Catalog, messages about the Import process are displayed in this tab.

Table 13-65 • Output Window Context Menus


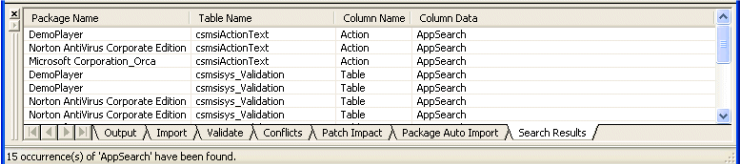
Tab	Description
Validate Tab	<p>After you perform validation, all of the Errors, Warnings, and Info messages that were generated are listed in table format. Each table row lists an icon to indicate whether it is an Error (✖), Warning (⚠), or Informational message (ℹ), the name of the ICE that generated it, and a brief description of what caused it to occur. To sort this list, click on the column heading of the column you want to sort by.</p>  <p>Note • If you used the <i>Validate Package</i> function to validate an MSI package that has been imported into the <i>Application Catalog</i>, these results are persisted and can be viewed on the <i>Validation View</i>. See Viewing ICE Error Information.</p>
Conflicts Tab	<p>After the Conflict Wizard is run to identify conflicts between a Windows Installer package and packages already cataloged in ConflictSolver, or if conflicts are checked during import, all of the Errors and Warnings that were generated are listed in a table format.</p> <p>Each table row lists an icon to indicate whether it is an Error (✖) or a Warning (⚠), the name of the ACE that generated it, and a brief description of what caused it to occur. To sort this list, click on the column heading of the column you want to sort by.</p> <p>If a row is active, you can double-click on it to view that row's associated table. The Tables View is launched and the table and/or table cells that are causing the problem are highlighted. If a row is grayed out, it indicates that the table cannot be viewed in ConflictSolver (perhaps because it is in an external package). To resolve conflicts, see Resolving Conflicts.</p>

Table 13-65 • Output Window Context Menus

Tab	Description
Patch Impact Tab	<p>After the Patch Impact Analysis Wizard is run to identify conflicts between a Microsoft patch and a Windows Installer package or OS Snapshot, all of the impacts that were generated are listed in a table format. Each table row lists the following information:</p> <ul style="list-style-type: none">● Patch—Name of the patch that caused the conflict.● Product—Name of the impacted product.● Type—The type of impact.● Description—Description of the conflict that was found between the patch and the product. <p>The following is an example of a conflict description:</p> <p>Package 'WindowsXP v5.00' uses version '6.0.9589' of file 'SystemFolder\expsrv.dll' but Patch 'Windows2000-KB837001-x86-ENU' uses version '6.0.72.9589' of the same file</p> <p>If you double-click on a row in this window, the conflicting file on the target product that is being impacted by the patch will be displayed and highlighted in the Tables View.</p> <p>For further information on the impacts generated by the Patch Impact Analysis, you may want to generate a Patch Report, or view the Product tab's Patch Impacts Analysis View, Dependencies View, and Associated Patches View. See Analyzing the Impact of Installing a Microsoft Operating System Patch for more information.</p>
Package Auto Import Tab	<p>When the Package Auto Import Wizard is run to automatically import packages in a Remote Application Catalog or Network Directory, messages about the import process are displayed in this tab.</p>
Search Results Tab	<p>When Find is used to search for data in Application Catalog tables, the data that is found is displayed in this tab, in the following format:</p> <div></div> <p>If you double click on this data, ConflictSolver will navigate to the appropriate record in the Tables View, and that record will be highlighted.</p>



Note • All Output Window tabs are shown in both the Application Manager View and the ConflictSolver View.

Views

The following views are associated with the Products tab in ConflictSolver:

- [Product View](#)
- [OS Snapshot View](#)
- [Other Setup Types View](#)
- [NCP Views](#)
- [Patches Tab Views](#)

Product View

When you select a Windows Installer Package (.msi) in the Product View, details about that product are displayed in the right pane of the user interface.

The following information is included:

Table 13-66 • Product View Information

Field	Description
Manufacturer	The manufacturer of the package.
Subscribed	If this package is associated with a subscribed database, the name of that database is listed here. If it is not, the following statement appears: Not associated with any subscribed database.
Conflict Results	If any persisted conflict data exists, this is a hyperlink to the package's Conflicts View.
Package Code	The globally unique identifier (GUID) for the setup package.
Product Code	A string that uniquely identifies the product.
Upgrade Code	A string used to upgrade the application. The upgrade code for a package groups that package into a specific product family.
Language	The language of this package.
File	This can be either a hard-coded path or a UNC path.
Imported On	The date and time the package was imported.
Original File	For patches, this is the original MSI package against which patch packages were applied to get to the final imported MSI package.
Transforms	This can be either a hard-coded path or a UNC path.

Table 13-66 • Product View Information (cont.)

Field	Description
Patches	The patch files that were used to get to the final imported MSI package.
Groups	Any groups to which the package belongs.
Description	You can edit this with additional information about the product.



Note • If the package and/or transforms are no longer in their original import directory, you can locate the file(s) from the provided hyperlink. You are also informed if the last modified date for the package in ConflictSolver does not match the last modified date of the package in its external location. You are given the opportunity to reimport the package to keep it synchronized in ConflictSolver.

If you click on the plus sign to expand a package in the Product View, a node is listed for each available constituent view. For example, for a Windows Installer package, the following nodes are listed:

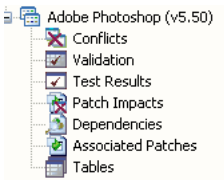


Figure 13-2: Product View Nodes

When you select one of these nodes, a constituent view opens in the right pane:

- Conflicts View
- Validation View
- Test Results View
- Patch Impact View (Products Tab)
- Dependencies View
- Associated Patches View
- Tables View



Note • The **Only Display View Nodes With Data** option on the **General** tab of the **ConflictSolver Options** dialog box controls whether product nodes (constituent views) appear if no data is contained in that view. If you select the option, products containing views without data will not display those views. For example, if a product has no Dependencies, then the Dependencies node is not displayed for that product.

Conflicts View

Each product contains a Conflicts view, where you can access information pertaining to the last execution of the Conflict Wizard. This persisted conflict data allows you to view when the last execution was performed, the products that it was run against, the ACE rules used and conflicts discovered. The view also provides a list of updated, deleted, or added products that may necessitate performing conflict identification again.

Summary Information

The main area of this view displays summary information for the last conflict identification execution. This information includes:

Table 13-67 • Conflicts View Summary Information

Item	Description
[Name of Package] Conflict View	The name of the product on which conflict identification was performed is listed above Conflict View in the title bar.
[Date and Time]	The date and time of the last conflict identification.
Total number of conflicts	The number of conflicts detected.
Automatically resolvable conflicts	The number of detected conflicts which can be resolved automatically.
Manually resolvable conflicts	The number of conflicts which can be resolved manually.
Resolved conflicts	The number of detected conflicts which have been resolved.
ACE rules evaluated	Click this link to open the ACE Rules tab of the Conflict Information Dialog Box with a list of the ACE rules used during conflict identification.
Packages Evaluated	Click this link to open the Products tab of the Conflict Information Dialog Box with a list of the packages that were included in conflict identification.

Also, the following information may also be listed:

- The number of checked packages that have changed since conflict identification.
- The number of checked packages that have been deleted from ConflictSolver since conflict identification.
- The number of packages that have been imported into ConflictSolver since conflict identification.

Links

When you click on one of the informational links, the [Conflict Information Dialog Box](#) opens, displaying information on the selected link. For example, if you click on the **ACE rules evaluated** link, the **Conflict Information** dialog box opens with the **ACE Rules** tab selected.

If you click on any of the conflict links, the list will be filtered by that type of conflict. For example, if you click on the **Automatically resolvable conflicts** link, only automatically resolvable conflicts will appear in the list.

Run Again and Reimport Buttons

Click **Run Again** to perform conflict identification on the package again, using the specified packages (if they still exist in ConflictSolver) and ACE rules. Because packages may have been removed or modified, conflict identification may return different results. Click **Reimport** to import the package into ConflictSolver again. This is usually done if you have already modified the package, and need to run conflict identification again.

Conflicts List

A list of the conflicts found during conflict analysis is displayed. You can filter this list by making selections from the list, such as **Show all conflicts**, **Show automatically resolvable conflicts**, etc.

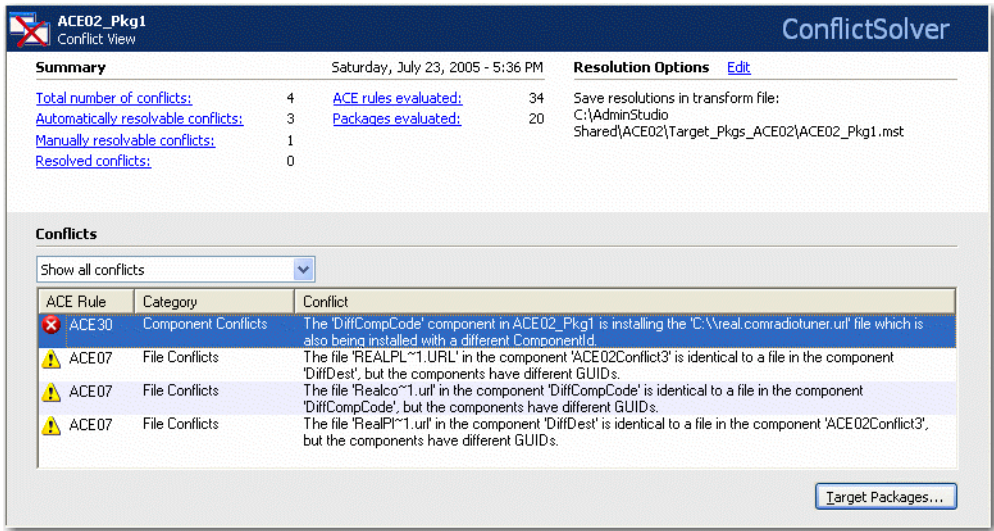


Figure 13-3: Conflict View

The icon displayed next to each conflict indicates its status:

Table 13-68 • Conflict Status Icons

Icon	Status
	Resolvable
	Resolved

Table 13-68 • Conflict Status Icons

Icon	Status
	Unresolvable

For each conflict, the ACE Rule number, the ACE category, and a description of the conflict is listed.

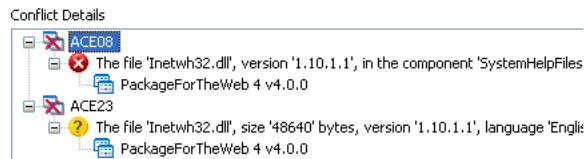
When evaluating conflicts between one source package and multiple destination packages, the same conflict may be detected between the source package and more than one destination package. These conflicts are consolidated and displayed as single entry. To see a list of all of the target packages that detected the same conflict, click the **Target Packages** button to view a list of the package on a pop-up dialog box.

In many cases, these errors can be resolved by clicking either **Resolve** or **Resolve All**. In cases where no automatic resolution is possible, the Unresolvable icon (✖) appears next to the conflict and the **Resolve** button is disabled.

If one of the following options on the [Resolution Tab](#) of the **Options** dialog box is selected, several additional dialog boxes could appear prompting for approval or additional information:

- If the **Preview and approve all resolutions** option is selected, the **Resolution Details** dialog box appears, listing the changes that are going to be made and prompting you to approve them.
- If the **Maximize user flexibility in selecting resolutions** option is selected, and if you are using a CARD with multiple resolution options, the **CARD Resolutions** dialog box would display a list of resolutions and would prompt you to select one.

For File Conflicts, click the plus sign next to the conflict icon to see the package that conflicts with the selected package.



Resolution Options

This section displays the current resolution option: either resolve in the original Windows Installer package, or using the specified transform. Click **Edit** to change the current settings on the [Resolution Options Dialog Box](#).

Validation View

The Validation View, which is displayed when you select the Validation node under a Windows Installer package on the Product View, lists all ICE Errors and Warnings that were generated the last time the selected package was validated. If a package has not been validated, no ICE errors will be listed.



Note • To validate a package that has already been imported into the Application Catalog, use the Validate Package function available on the Product View.

ICE errors include an icon to indicate whether it is an Error (❌) or a Warning (⚠️), the name of the ICE that generated it, and a brief Description of what caused it to occur:

Adobe Acrobat 6.0 Professional		
Validation View		
View:	All	Sunday, August 08, 2004
	ICE	Description
❌	ICE03	Invalid Filename; Table: File, Column: FileName, Key(s): esdupdate.dll
⚠️	ICE09	Component: Distiller_Plug_in_Roman_Sys is a non-permanent system component
⚠️	ICE09	Component: PIPRes.dll.cht is a non-permanent system component
⚠️	ICE09	Component: PIPRes.dll.fr is a non-permanent system component
⚠️	ICE09	Component: PIPRes.dll.ger is a non-permanent system component
⚠️	ICE09	Component: PIPRes.dll.jpn is a non-permanent system component
⚠️	ICE09	Component: PIPRes.dll.kor is a non-permanent system component

Figure 13-4: Example of ICE Error Listing

You can sort by any column by clicking the appropriate column title.

Viewing ICE Error Information

To easily view the full text of the ICE error description, select an ICE Error or Warning number and then select Properties from the context menu. The Properties Dialog Box opens listing the ICE error number and its full description. For more information, see [Viewing ICE Error Information](#).

Suppressing an ICE Error

If you do not want ConflictSolver to check for a particular ICE error during subsequent validations of a package, you can choose to suppress an ICE error. See [Suppressing an ICE Error](#) for more information. Suppressed ICE errors are listed in gray:

	ICE	Description
❌	ICE03	Invalid Filename; Table: File, Column: FileName, Key(s): esdupdate.dll
⚠️	ICE09	Component: Distiller_Win_NT_System_files is a non-permanent system component
⚠️	ICE09	Component: Distiller_Win9X_System_files is a non-permanent system component

Figure 13-5: ICE Error Listing with a Suppressed ICE Error (in gray)

You can make a selection from the View list to filter the list by Enabled or Suppressed status.

Deleting an ICE Error

You can permanently delete an ICE error from subsequent execution and reporting for that package. ICE errors that are deleted are no longer listed in the Validation View and will not be listed in the Package Report. See [Deleting an ICE Error](#) for more information.

Test Results View

You can select a package in ConflictSolver and then choose to test the package using PackageExpert. Test results will be displayed in the ConflictSolver **Test Results** view.

For more information, see [Testing Packages From ConflictSolver](#) and [Viewing Test Results in ConflictSolver](#).

Patch Impact View (Products Tab)

The **Patch Impact View** is displayed when you select the **Patch Impacts** node under a product on the **Products** tab.

Summary View

The information displayed on the **Patch Impact View** is dependent upon the selection made in the **Impact category** list. The **Summary View**, which is displayed when **Summary** is selected from the **Impact category list**, displays a list of patches for which there is patch impact data persisted against the product. The following information is displayed:

Table 13-69 • Summary View Information

Option	Description
Id	Number identifying this patch's associated Microsoft Security Bulletin.
Name	Name of the patch file.
Title	Title of the patch.
Release Date	Date this patch was published by Microsoft.

File Impacts View

The **File Impacts View**, which is displayed when **File Impacts** is selected from the **Impact category** list and you have file impacts persisted, lists all impacts against this product or OS Snapshot and identifies the patch that caused the impact. If you double-click on one of the patches, the Patch View for that patch will open. The following information is displayed:

Table 13-70 • File Impacts View Information

Option	Description
Description	Description of the impact.

Table 13-70 • File Impacts View Information

Option	Description
Id	Number identifying this patch's associated Microsoft Security Bulletin.
Name	Name of the patch file.
Title	Title of the patch.



Note • The **Patch Impact View** on the *Products* tab is a corollary to the [Impact Analysis View \(Patches Tab\)](#).

Dependencies View

On the Dependencies View, which is accessed by selecting the Dependencies node under a product in the Product View, you can view a list of all of the files of a selected product that have dependencies with files used by other products or operating systems in the Application Catalog. This view is displayed for Windows Installer .msi packages in which file dependency information exists.

Dependency information is generated by the Patch Impact Analysis Wizard. You can also generate this information by selecting a product in the Product View and then selecting Scan for Dependencies from the context menu.



Note • If the **Only Display View Nodes With Data** option on the ConflictSolver Options Dialog **General** tab is selected, if you scan for dependencies and no dependencies are found, the **Dependencies** node will still not be displayed.

The following information is displayed:

Table 13-71 • Dependencies View Information

Option	Description
Files With Dependencies	Make selections from this list to further filter the output listing, or select (All) to display all dependencies.
File	Name of a file included in this product that is dependent upon a file used by another product or operating system in the Application Catalog.
Dependent File	Name of the file that this product's file is dependent upon.
Size	Size of the dependent file. This field is populated for file dependencies identified from the MSI package itself. For dependencies that were derived from the static scanning process, this field will be undefined.

Table 13-71 • Dependencies View Information (cont.)

Option	Description
Version	Version of the dependent file. For dependencies that were derived from the static scanning process, this field will be undefined.
Language	Language of the dependent file. For dependencies that were derived from the static scanning process, this field will be undefined.

Associated Patches View

On the Associated Patches View, you can view a list of imported patches that, if installed, would update the selected product. ConflictSolver examines the patches in the catalog and attempts to identify those patches which will impact this package.

Due to differences in the way versions are compared, it is possible that other patches that impact this package may exist. For more definitive information, open the **Patch Properties** dialog box and compare the product and OS snapshot version information of the patch against the specific product and version information.

In the Associated Patches View, the following information is displayed:

Table 13-72 • Associated Patches View Information

Option	Description
Name	Name of the patch that is associated with this product.
Information	Microsoft Security Bulletin identification number and description of the patch.

If you double-click on a patch in the Associated Patches View, the Patch View (on the Patches tab) for that patch opens, listing general information on the selected patch.

Tables View

The Tables view provides a way to view the data for a given package in the Application Catalog. Select the specific table you want to view from the Tables list at the top of the view.

Most tables are derived directly from standard MSI tables, as described in the Windows Installer SDK online help.

When building your own ACE rules to use for conflict identification, it is important to understand the data available for packages so you can construct the necessary rule.

OS Snapshot View

When you click on an OS Snapshot in the Product View, details about the snapshot appear in the right pane of the user interface.

The following information is displayed:

Table 13-73 • OS Snapshot View Information

Field	Description
Version	Version of the operating system of the OS Snapshot, such as Windows XP - 5.1.2600.
Language	The language the operating system was written for.
File	This can be either a hard-coded path or a UNC path.
Imported On	The date and time the OS Snapshot was imported.
Description	You can edit this with additional information about the OS Snapshot.

Click the plus sign next to the OS Snapshot icon to view these OS Snapshot constituent views:

- [Tables View for OS Snapshots](#)

Tables View for OS Snapshots

The Tables View for an OS Snapshot is identical to the Tables View shown when a Product is selected. See [Tables View](#) in the Product View section.

Other Setup Types View

When you click on an Other Setup Types package (a package containing non-MSI based setup files) in the Product View, details about that setup are displayed in the right pane of the user interface.

The following information is included:

Table 13-74 • Other Setup Types View Information

Field	Description
Main Directory	The original location of the non-MSI based setup files when they were imported into ConflictSolver.
Imported On	The date and time the setup was imported.

Table 13-74 • Other Setup Types View Information

Field	Description
Files	<p>A listing of the individual files making up the setup are listed, allowing you to view or delete files by making a selection from the context menu.</p> <ul style="list-style-type: none"> • To view the contents of a file, select View from the context menu. You can only view files supported by applications installed on your workstation. If you attempt to view a non-supported file, no View window will appear. If you attempt to view an executable, that executable will be launched. • To delete a file from this setup, select Delete from the context menu.

If you expand an Other Setup Type, you can view its constituent view, the [Tables View for Other Setup Types](#).

Tables View for Other Setup Types

The Tables View for an Other Setup Types package is identical to the Tables View shown when a Product is selected. See [Tables View](#) in the Product View section.

NCP Views

When you select a Marimba NCP file (.ncp) from the Product View, details about that file are displayed in the right pane of the user interface.

The following information is included:

Table 13-75 • NCP Views Information

Column	Description
Conflict Results	Date that this package was last checked for conflicts. Click the link to jump to the Conflicts View for Marimba NCP Files .
Version	Version of the software package
File	Name and path of the NCP file that was imported
Imported On	Date when the file was imported
Description	After the NCP file is imported, you can edit this with additional information.

If you expand an NCP file, you can view its constituent views:

- [Conflicts View for Marimba NCP Files](#)
- [Tables View for Marimba NCP Files](#)

Conflicts View for Marimba NCP Files

The Conflicts View for an NCP File is identical to the Conflicts View shown when a Product is selected. See [Conflicts View](#) in the Product View section.

Tables View for Marimba NCP Files

The Tables View for an NCP file is identical to the Tables View shown when a Product is selected. See [Tables View](#) in the Product View section.

Patches Tab Views

The Patches tab includes the following views which provide content and impact information about Microsoft patches that have been imported into the Application Catalog.

Table 13-76 • Patches Tab Views

View	Description
Patches Group View	Opens when the root group in the Patches tab is selected, and lists all of the groups that have been created in the Patches tab.
New Patches Group View	All new patches are imported into the New Patches group, and this view lists all of the patches in that group.
Group View of a Selected Group	Opens when a group other than the root group in the Patches tab is selected. For each selected group, a list of all of the patches in that group is displayed.
Patch View	Lists general content information on a selected patch.
Impact Analysis View (Patches Tab)	Opens when the Impacts node under a Patch on the Patches tab is selected. Displays a list of patches for which there is patch impact data persisted against a product. It also lists all impacts against a product, and identifies the patch that caused the impact.

Our patches tree structure would then look and operate much the same as our Group - Package tree including a RC menu. Drag-Drop support would be provided for the groups and patches in much the same way as for groups and products.

Patches Group View

The Patches Group View opens when the root group in the **Patches** tab is selected. The Patches Group View lists all groups that have been created in the Patches tab.

All new patches are imported into the **New Patches** group, and then you can organize the patches into other groups according to your business needs. See [Organizing Your Application Catalog Using Groups](#).

The **New Patches** group is automatically created during installation. While it can be renamed, it cannot be deleted.

Context Menu Options

When the root group in the Patches Tab is selected, the following items are available on the context menu:

- **Refresh**—Refresh the patch listing to reflect the most recent modifications.
- **Import Patches**—Opens the OS Security Patch Wizard interface so that you can import patches into the Application Catalog.
- **New Group**—Create a new group.
- **Rename**—Rename the selected group.
- **Properties**—Open the **Group Properties** dialog box.

New Patches Group View

The New Patches Group View opens when the **New Patches** group on the Application Manager or ConflictSolver **Patches** tab is selected. All new patches are imported into the **New Patches** group. You can then organize the patches into other groups according to your business needs. See [Organizing Your Application Catalog Using Groups](#).

The **New Patches** group is automatically created during installation. While it can be renamed, it cannot be deleted.

The New Patches Group View displays a list of all of the patches in that group, including the following information:

Table 13-77 • New Patches Group View Information

Option	Description
Name	Name of patch file.
Description	Description of the patch file.
Release Date	Date the patch was released by Microsoft.
Import Date	Date the patch was imported into the Application Catalog.

If you select a patch in this list, detailed patch properties are displayed in the area to the left of the list, including the **Title**, **Summary**, and **Release Date** of the patch.

Context Menu Options

When the **New Patches** group in the **Patches Tab** is selected, the following items are available on the context menu:

- **New Group**—Create a new group.

- **Rename**—Rename the selected group.
- **Properties**—Open the **Group Properties** dialog box.

Group View of a Selected Group

The Group View of a selected group opens when a group other than the root group or the **New Patches** group in the **Patches** tab is selected. For each selected group, a list of all of the patches in that group is displayed, including the following information:

Table 13-78 • Group View of a Selected Group Information

Option	Description
Name	Name of patch file.
Description	Description of the patch file.
Release Date	Date the patch was released by Microsoft.
Import Date	Date the patch was imported into the Application Catalog.

If you select a patch in this list, detailed patch properties are displayed in the area to the left of the list, including the **Title**, **Summary**, and **Release Date** of the patch.

Context Menu Options

When a group other than the root group or the **New Patches** group in the **Patches Tab** is selected, the following items are available on the context menu:

- **New Group**—Create a new group.
- **Rename**—Rename the selected group.
- **Delete**—Delete the selected group.
- **Cut**—Copy the selected group to the clipboard so that you can remove it from its current location and paste it into another location.
- **Paste**—Insert the patch or the group on the clipboard into the selected location.
- **Properties**—Open the **Group Properties** dialog box.

Patch View

The **Patch View**, which is displayed when a patch is selected on the **Patches** tab, lists general information on the selected patch. The following information is included:

Table 13-79 • Patch View Information

Option	Description
ID	Microsoft Security Bulletin ID. Click this link to view this bulletin on the Microsoft Web site.
Title	Title of patch.
Release Date	Date the patch was released by Microsoft.
KB Article	Microsoft Knowledge Base article ID. Click this link to view this article on the Microsoft Web site.
Imported On	Date patch was imported into the Application Catalog
Groups	List of all of the groups that this patch is included in.
Description	You can enter a description of the patch in this field.

In the Application Manager, you can view additional detailed patch information by selecting a patch and then selecting **Properties** from the context menu.

Context Menu Options

When a patch in the **Patches Tab** is selected, the following items are available on the context menu:

- **Rename**—Rename the selected patch.
- **Delete**—Delete the selected patch.
- **Cut**—Copy the selected patch to the clipboard so that you can remove it from its current location and paste it into another location.
- **Copy**—Copy the selected patch to the clipboard so that you can paste a copy of it into another location.
- **Paste**—Insert the patch on the clipboard into the selected location.
- **Generate Report**—Generate a Patch Impact Analysis Report for that patch. See [Generating the Patch Report](#).
- **Properties**—Open the **Patch Properties** dialog box for that patch.

Impact Analysis View (Patches Tab)

The **Impact Analysis View** opens when you select the **Impacts** node under a patch on the ConflictSolver **Patches** tab. The information displayed on this view is dependent upon the selection made in the **Impact category** list.

Summary View

The **Summary View**, which is displayed when **Summary** is selected from the **Impact category** list, lists details regarding the last time impact analysis was performed, including:

- **Products that were checked for impacts** for the selected patch during patch impact analysis.
- **Products that have been updated** since the last impact analysis was performed.
- **Products that have been removed** from the Application Catalog since the last impact analysis was performed.
- **Products that have been added** to the Application Catalog since the last impact analysis was performed.

If one of these four categories include one or more products, a link to the [Patch Impact Information Dialog Box](#) is provided, which lists information on individual tabs for each of the categories. If a category does not include any products, its tab is not displayed.

Click **Run Again** to run patch impact analysis based upon the selections that were previously chosen in the Patch Impact Analysis Wizard. Analysis messages are listed in the **Output** tab of the **Output Window**. When analysis is complete, patch conflicts are listed on the **Patch Impact** tab in table format. See [Analyzing the Impact of Installing a Microsoft Operating System Patch](#) for more information.

File Impacts View

On the **File Impacts View**, which is displayed when **File Impacts** is selected from the **Impact category** list, all impacts against this patch are listed, and the product(s) that the patch impacted are identified. The following information is displayed:

Table 13-80 • File Impacts View Information

Option	Description
Description	Description of the impact.
Package Name	Name of the product that this patch impacted.

If you double-click on one of the listed products, that product will be opened in the [Tables View](#) with the row that is causing the problem highlighted.



Note • The **Impacts Analysis View** on the **Patches** tab is a corollary to the [Patch Impact View \(Products Tab\)](#).

Dialog Boxes

The following dialog boxes are accessible from ConflictSolver:

- [About ConflictSolver Dialog Box](#)
- [ACE Rule Properties Dialog Box](#)
- [Add Ignore Table Dialog Box](#)
- [CARD Resolution Options Dialog Box](#)
- [Conflict Information Dialog Box](#)
- [Conflict Resolution Dialog Box](#)
- [Customize Dialog Box](#)
- [Expression Builder Dialog Box](#)
- [Options Dialog Box](#)
- [Patch Impact Information Dialog Box](#)
- [Resolution Details Dialog Box](#)
- [Resolution Operations Dialog Box](#)
- [Resolution Options Dialog Box](#)
- [Rules Viewer Dialog Box](#)

About ConflictSolver Dialog Box

The About InstallShield ConflictSolver dialog box can be accessed by selecting About ConflictSolver from the Help menu. This dialog box displays information about the product, including the full version number (essential if you need technical support).

ACE Rule Properties Dialog Box

The ACE Rule Properties dialog box allows you to edit an existing user-defined ACE rule. You can display the dialog by clicking Edit on the on the Rules Viewer dialog box.

The following tabs are part of the ACE Rule Properties dialog box:

- [General Information Tab](#)
- [Additional Information Tab](#)
- [Custom Options Tab](#)
- [Where Clause Tab](#)
- [DLL Information Tab](#)



Note • You can only edit user-defined ACE rules; you are not permitted to edit the ACE rules that were installed with ConflictSolver.

General Information Tab

From the General Information tab, you can configure information about the new ACE rule.

This information is used primarily for display information (Name, Brief Description, Description, and Information URL).

Table 13-81 • General Information Tab Information

Option	Description
Name	The name of the ACE, used to organize the rule in ConflictSolver. This is displayed in several places, including the Output Window, the Rules Viewer dialog box, and the Conflicts tab of the Options dialog box.
Associated Table	Select the table in the Application Catalog which will be queried in the user-defined ACE. This also determines which columns are available in the Expression Builder dialog box, and which tokens are available for the Error and Display strings on the Custom Options Panel of the Rules Wizard.
Brief Description	Enter a brief description which will be displayed in the Rules Viewer dialog box, the Conflicts tab of the Options dialog box, and in the Output Window. This description should be clear enough so users can understand when to use this ACE.
Description	Enter a description of the ACE, which is displayed at the bottom of the Rules Viewer dialog box when the ACE is selected and in the Output Window during conflict identification when the ACE executes.
Information URL	Provide a URL to get further information for the ACE. This URL appears in the Conflict Details area of the Conflicts View after conflicts have been identified.

To continue editing ACE Properties, click the Additional Information, Custom Options, Where Clause, or DLL Information tabs. To save your edits and close this dialog box, click OK.

Additional Information Tab

From the Additional Information tab, you can edit information for categorizing the ACE in relation to other ACEs in ConflictSolver.

Table 13-82 • Additional Information Tab Information

Option	Description
Category	Either select an existing category for this new rule, or enter the name for a new category. These categories are displayed in the Conflict View and the Conflicts tab of the Options dialog box. Ideally, any user-defined ACEs should be put in their own category.
Rule Type	<p>The Rule Type of this ACE is displayed (read only).</p> <ul style="list-style-type: none"> • Custom - Source Only Packages ACEs allow you to quickly test any column or any value of a table. For example, you could use a user-defined ACE to identify packages that create a desktop icon. To define a user-defined ACE, you use an SQL “Where” clause. • Custom - Source and Target Packages ACEs allow you to compare columns or values of Source package tables (new packages that you want to install onto a user’s system) to columns or values of Target package tables (packages already installed on a user’s system). For example, you could use a Source and Target Packages ACE to determine if the installation of a Source package onto a Target system would produce duplicate registry entries. To define a Source and Target Packages ACE, you must define an SQL “Where” clause, and specify a Join Column (a table column in the Application Catalog database that has a matching value for both the Source and Target packages). • DLL - User Provided DLL based ACEs allow you to run more complex tests—testing many tables in any combination. For example, you could use a DLL-Based ACE to confirm that a source product language is the same as all target product languages. To define a DLL-Based ACE, you use SQL and other programming commands. With DLL-Based ACEs, you can use a Conflict Application Resolution Definitions (CARDS) to fix the conflict.



To continue editing ACE Properties, click the General Information, Custom Options, Where Clause, or DLL Information tabs. To save your edits and close this dialog box, click OK.

Custom Options Tab

From the Custom Options tab, you can edit this ACE's display strings for the Output window and Conflict Details.

The following options are included:

Table 13-83 • Custom Options Tab Properties

Option	Description
Error String	<p>This string appears in the Output window when a violation of this ACE rule is detected during conflict identification. For example, if you were creating a user-defined ACE to identify packages that create a desktop icon, you could enter the following in this field:</p> <p>Failure in creating desktop icon</p>  <p>Note • Tokens allow you to insert values at run-time from the installation package into the string, such as specifying a file name. To use token replacement in the error string, use the arrow to the right of the Error String field. For more information, see Token Grammar.</p>
Display String	<p>This string appears in the Conflict Details area of the Conflicts View after conflicts have been identified. For example, if you were creating a user-defined ACE to identify packages that create a desktop icon, you could enter the following in this field:</p> <p>Duplicate desktop icon found</p>  <p>Note • Tokens allow you to insert values at run-time from the installation package into the string, such as specifying a file name. To use token replacement in the error string, use the arrow to the right of the Display String field. For more information, see Token Grammar.</p>
Severity	<p>Specify whether this ACE should be an Error or a Warning.</p>
Report 'No' results	<p>User-defined ACEs report conflicts based on the provided query. However, you may want to report the absence of the data if it could not be found. If you select this option, if the ACE does not return any results, it will be reported as an error (or warning), with the description and error strings as specified. If you expect a No result, do not use tokens in your display or error strings.</p>

To continue editing ACE Properties, click the General Information, Additional Information, or Where Clause tabs. To save your edits and close this dialog box, click OK.

Where Clause Tab

From the Where Clause tab, you can edit the Where clause for the ACE. If you do not know how to build a Where clause, you can click the Build Expressions button to launch the **Expression Builder** dialog box. You can also click Test to validate the Where clause syntax.

If you selected Custom - Source and Target Packages when you created this ACE, you must have also selected a Join Column—a table column in the Application Catalog database that has a matching value for both the Source and Target packages. Rows in each of the packages that have a matching value in the Join Column are selected and those rows are checked against the Source and Target Packages. For example, if you wanted to evaluate records from two tables that have a installation directory of C:\ProgramFiles, then you would specify Directory as the Join column. To change the Join Column, select a different column name from the list.

To continue editing ACE Properties, click the General Information, Additional Information, or Custom Options tabs. To save your edits and close this dialog box, click OK.



Tip • To improve query performance, enclose table names in square brackets ([]).

DLL Information Tab

From the **DLL Information** tab, you can edit specific information about the ACE/CARD DLL file and the ACE and CARD Function Names that DLL-based ACEs require to operate.

The following options are included:

Table 13-84 • DLL Information Tab Information

Option	Description
ACE/CARD DLL File	Select the name of the ACE DLL that you are testing.
ACE Function Name	Enter the name that you chose to “export” for this ACE function.
CARD Function Name	Enter the name that you chose to “export” for this CARD function.
Test	Click the Test button next to the ACE Function Name or CARD Function name to validate that the exported function does exist.

To continue editing ACE Properties, click the General Information or Additional Information tabs. To save your edits and close this dialog box, click OK.

Add Ignore Table Dialog Box

This dialog box allows you to specify a custom table to ignore during import of a package into ConflictSolver. You can also provide comments about the table.

Ignored tables will not automatically be imported when using the Import Custom Tables option in the Import Tab of the Options dialog box.

CARD Resolution Options Dialog Box

Some CARDS can resolve a conflict in more than one way. For example, ACE06 checks whether the executable module within the component is the key file, and accumulates a list of potential files to be used for the key file. Any one of these files could be designated as the key file to resolve the conflict.

If a CARD can be resolved in multiple ways, and if the Maximize user flexibility in selecting resolutions option is selected on the [Resolution Tab](#) of the **Options** dialog box, the CARD Resolution Options dialog box opens, prompting you to make a selection to determine how conflict resolution will proceed.

Conflict Information Dialog Box

When you click on one of the informational links in the Summary section of either the [Conflicts View](#) or [Conflict Resolution Dialog Box](#), the Conflict Information dialog box opens, displaying information on the selected link. For example, if you click on the ACE rule link, the Conflict Information dialog box opens with the ACE Rules tab selected.

The dialog box contains the following tabs (which are only displayed if they have data):

Table 13-85 • Conflict Information Dialog Box Tabs

Tab	Description
Products	The product(s) against which conflict identification was performed. The name, description, and any groups the products belong to are displayed.
ACE Rules	The category and description for each ACE rule used in conflict identification.
Updated Products	The name, description, and modification date of any relevant product that has been updated since the last conflict identification was performed.
Deleted Products	The name and description of any relevant product that has been removed from ConflictSolver since the last conflict identification was performed.
New Products	The name, description, and import date of any product that has been added to ConflictSolver since the last conflict identification was performed.



Note • This dialog box is similar in function to the Patch Impact Information dialog box.

Conflict Resolution Dialog Box

You can display the **Conflict Resolution** dialog box either by running the Conflict Wizard and selecting the **Run Resolution** option in the **Conflict Rules** panel, or by selecting **Resolve External Package Conflicts** from the **Conflicts** menu after a previous conflict identification between an external package and products in ConflictSolver.



Tip • *It is strongly recommended that you import packages into ConflictSolver prior to performing conflict identification, instead of running conflicts between an external package and products in ConflictSolver.*

From this dialog box, you can access information pertaining to the last execution of the Conflict Wizard. This persisted conflict data allows you to view when the last execution was performed, the products that it was run against, the ACE rules used, and conflicts discovered. The view also provides a list of updated, deleted, or added products that may necessitate performing conflict identification again.

Summary Information

The main area of this view displays summary information for the last conflict identification execution. This includes:

- The name of the product on which conflict identification was performed.
- The date and time of the last conflict identification.
- The number of conflicts detected.
- The number of detected conflicts which can be resolved automatically.
- The products the current product was checked against.
- The ACE rules used during conflict identification.
- The number of checked packages that have changed since conflict identification.
- The number of checked packages that have been deleted from ConflictSolver since conflict identification.
- The number of packages that have been imported into ConflictSolver since conflict identification.

Links

When you click on one of the informational links, the [Conflict Information Dialog Box](#) appears, displaying information on the selected link. For example, if you click on the ACE rule link, the Conflict Information dialog box opens with the ACE Rules tab selected.




Run Again

Click Run Again to perform conflict identification on the package again, using the specified packages (if they still exist in ConflictSolver) and ACE rules. Because packages may have been removed or modified, conflict identification may return different results.

Conflict Information

Using the pull-down list at the top of the view, you can toggle between summary information and conflict-specific information. When you select a specific conflict type, the view changes to display a list of corresponding errors in the product, grouped by ACE. The icon displayed next to the conflict indicates its status:

Table 13-86 • Conflict Status Icons

Icon	Status
	Resolvable
	Resolved
	Unresolvable

In many cases, these errors can be resolved by clicking either Resolve or Resolve All. In cases where no automatic resolution is possible, the Unresolvable icon (✖) appears next to the conflict and the Resolve button is disabled.

If one of the following options on the Resolution Tab of the Options Dialog Box is selected, several additional dialog boxes could appear prompting for approval or additional information:

- If the **Preview and approve all resolutions** option is selected, the [Resolution Details Dialog Box](#) appears, listing the changes that are going to be made and prompting you to approve them.
- If the **Maximize user flexibility in selecting resolutions** option is selected, and if you are using a CARD with multiple resolution options, a list of options from you to choose from could appear the [CARD Resolution Options Dialog Box](#), and you would be prompted to select one.

For File Conflicts, click the plus sign next to the conflict icon to see the package that conflicts with the selected package.

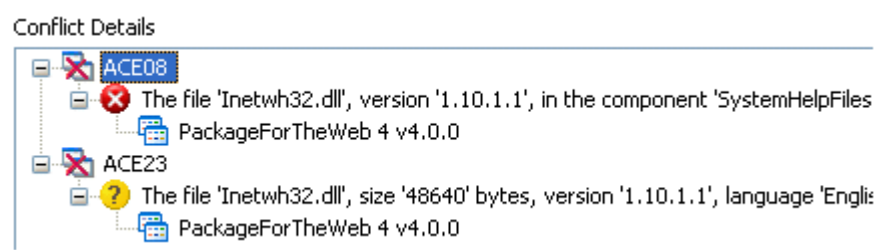


Figure 13-6: Conflicts listed in the Conflicts View

Resolution Options

This section displays the current resolution option: either resolve in the original Windows Installer package, or using the specified transform. Click Edit to change the current settings on the Resolution Options dialog box.

Customize Dialog Box

The Customize Dialog box, which is accessed by selecting Customize from the Tools menu, allows you to customize which tool bars are available in the ConflictSolver user interface, as well as the buttons that are available on the tool bars. The dialog box consists of two tab panels:

Toolbars

From the Toolbars panel, you can select viewing properties for all tool bars, such as whether teletypes are displayed, the style of the toolbar, and the size of the buttons. You can also create your own custom toolbar, onto which you can place buttons found in the Command tab.

Command

The Command panel allows you to customize tool bars and the menu bar. Drag the command or menu you want to add to the existing toolbar; it appears where you place it. To remove a command or menu, select it and drag it off the toolbar.

Properties Dialog Box

To open the Properties dialog box, select an ICE on the Validation View and select Properties from the context menu. The full Description of the ICE is displayed, along with the ICE number. You can add a comment in the Explanation text box.

Changing the State of an ICE Error




If you do not want ConflictSolver to check for a particular ICE error during subsequent validations of a package, you can choose to suppress it. You might choose to suppress an ICE error that is a known issue at your organization which does not need additional corrections. The ICE Error state is listed in the State area of the Properties dialog box:

- **Enabled**—Continue to check this ICE rule during subsequent validations of this package.
- **Disabled**—Do not check this ICE during subsequent validations of this package.

You can change the State of an ICE error by selecting **Enabled** or **Suppressed** on Properties Dialog Box, or by selecting the ICE error on the Validation View and then selecting Enable or Suppress from the context menu.

If you choose to suppress an ICE, you could document the reason for the suppression in the Explanation text box on the Properties Dialog Box.

On the Validation View, suppressed ICE errors are listed in gray:

	ICE	Description
	ICE03	Invalid Filename; Table: File, Column: FileName, Key(s): esdupdate.dll
	ICE09	Component: Distiller_Win_NT_System_files is a non-permanent system component
	ICE09	Component: Distiller_Win9X_System_files is a non-permanent system component




Note • Suppressed ICE errors are still listed in the Package Report for that package. To remove an ICE error from the Package Report, you need to Delete it rather than Suppress it. See [Deleting an ICE Error](#) for more information.

Expression Builder Dialog Box

The Expression Builder dialog box, available by clicking Build Expression on the **Where Clause Panel** in the Rules Wizard (when creating a new user-defined ACE) or from the Where Clause Tab of the [ACE Rule Properties Dialog Box](#) (when editing a user-defined ACE), allows you to build simple Where clause expressions for ConflictSolver user-defined ACEs.

Set values for the following options:

Table 13-87 • Expression Builder Dialog Box Properties

Option	Description
Table Columns	This list is populated from the table columns in the table defined in the Rules Wizard General Information Panel. Select the table column used in this Where clause.
Comparison Operator	<p>Pick an operator to use for comparison in the Where clause. You can pick from the following:</p> <ul style="list-style-type: none"> • = (Equal To) • <> (Not Equal To) • > (Greater Than) • < (Less Than) • >= (Greater Than or Equal To) • <= (Less Than or Equal To)
Constant	<p>This constant can be a numerical value or string value. The property label will change based on the expected constant type. This value is compared against the data in the specified table.</p>  <p>Note • When using the Expression Builder dialog box to create a Source and Target Packages custom ACE to compare the value of a column in the source table to the value of a column in the target table, you can select the first table column name from the Table Columns list. However, you have to manually enter the second table column name in the Constant text box. When doing so, enter the table column name using the same syntax that is used in the Table Columns list: [Source].[ColumnName] or [Target].[ColumnName]. See Creating a Custom/Source and Target Packages ACE.</p>
Expression Operator	If there is more than one expression in the Where clause, you can specify an operator to join the current expression to the previous expression.



Tip • When you are constructing simple expressions, it is helpful to use the Expression Builder dialog box, but you are not limited to the formatting options that the Expression Builder provides to you. If you know how to write Where clauses in SQL, you can use significantly more powerful expressions by entering them directly in the Where Clause text box on the Where Clause Panel of the Rules Wizard or on the Where Clause Tab of the [ACE Rule Properties Dialog Box](#).

Options Dialog Box

The Options dialog box, accessible by selecting Options from the Tools menu, provides a way to configure how ConflictSolver handles validation and conflict identification, as well as some general settings. This dialog box is divided into the following tabs:

- [Rules Tab](#)
- [Duplicate Package Tab](#)
- [Extended Attributes Tab](#)
- [General Tab](#)
- [Import Tab](#)
- [Resolution Tab](#)
- [Validate Tab](#)
- [History Tab](#)


Rules Tab

From the **Rules** tab of the Options dialog box, you can configure the following options:

Table 13-88 • Rules Tab Properties

Option	Description
Suppress Warnings/Info	When these boxes are checked, the messages of that type will not be displayed in the Conflicts tab of the Output Window during conflict identification. It is strongly recommended you do not suppress warning messages so you can ensure proper conflict identification.

Table 13-88 • Rules Tab Properties (cont.)

Option	Description
Check conflicts across all source and target packages	<p>When performing conflict analysis using multiple source packages and one or more target packages, ConflictSolver will evaluate each source package against each target package.</p> <p>However, if you want ConflictSolver to also perform conflict analysis of each source package against every other source package and each target package against every other target package, select this option.</p>  <p>Note • If performing conflict analysis with multiple source packages, and if this option is selected, both ACE21 and ACE22 are automatically evaluated when either one is selected. If only one of them is evaluated, then there is the possibility that conflicts will not be detected.</p>
Enabled Predefined and Custom Rules	<p>There are two types of ACE rules:</p> <ul style="list-style-type: none"> • Conflict ACEs—Detect conflicts between two Windows Installer packages. • Best Practice ACEs—Internally perform checks against the structure of a Windows Installer package. (Similar to Microsoft’s ICE validation rules.) <p>Select one of these radio buttons to toggle the ACE tree listing between Conflict ACEs and Best Practice ACEs:</p> <ul style="list-style-type: none"> • Show best practice rules • Show conflict detection rules <p>In both the Conflict ACE and Best Practice ACE trees, if you want a specific type of conflict identified by default, select the appropriate check box.</p> <p>ACEs associated with unselected boxes will not be performed by default during conflict identification. However, you can override these in the Summary Panel of the Conflict Wizard.</p>
Custom ACE Rule File	<p>The user-defined ACE file specified here is run after the pre-defined ACE rules are run. The selection of this user-defined ACE file will affect the default Conflict Types displayed on this dialog (described above), as well as those displayed on the Rules Viewer and the Conflict Wizard.</p> <p>By default, a file path to an initially empty user-defined ACE file is provided for you. If you have already created a user-defined ACE, specify the location of that user-defined ACE file to activate it. Only one user-defined ACE file can be active at one time.</p> <p>You use user-defined ACEs to extend the functionality of pre-defined ACEs with company-specific functionality. By selecting different user-defined ACE files, you can organize rules appropriate for individual users in your organization. See User-Defined ACEs for more information.</p>

Duplicate Package Tab

When you import a package into the ConflictSolver Application Catalog database, ConflictSolver checks specific identifiers that are selected on the Duplicate Package Tab to determine if that package has already been imported.

If ConflictSolver determines that you are attempting to import a duplicate package (based upon the selected identifiers), the Product Name Change Dialog Box appears, where you are given the opportunity to enter a name of your choosing to identify the duplicate package before the package is imported.




Note • If ConflictSolver is performing a bulk import or reimport, it still identifies duplicate packages using the user-specified criterion. However, ConflictSolver will generate a unique displayed product name and allow the import to proceed without prompting the user.

The identifiers you can select on the Duplicate Package Tab are as follows:

Table 13-89 • Duplicate Package Tab Properties

Option	Description
PackageCode Property	Identifier of package product was installed from. No two non-identical .msi files should ever have the same package code.
ProductCode Property	Unique identifier for the particular product release, represented as a string GUID, for example {12345678-1234-1234-1234-123456789012}.
Product Language	The language the installer should use for any strings in the user interface that are not authored into the database.
ProductVersion	Version of the product in string format. The format of the string is: major.minor.build.
List of Transform Files	A list of the transformations associated with this package import operation.
[None Selected]	If you do not select any of these five identifiers, ConflictSolver checks the ProductName Property identifier to determine if a package is a duplicate.

Table 13-89 • Duplicate Package Tab Properties

Option	Description
Duplicate Package Naming Syntax	<p>When it identifies a duplicate package, Application Manager generates a new name for that package using the syntax specified in this field. The default syntax is:</p> <p>[Manufacturer]_[ProductName]</p> <p>This means that if Application Manager encountered a duplicate package, it would prepend the duplicate's Product Name with the Manufacturer's name and, if necessary, append the Product Name with numbers. For example, the second time that PowerPoint is imported, its name would be changed to:</p> <p>Microsoft Corporation_PowerPoint</p> <p>To change the default naming syntax, edit this field.</p> <p>This generated name is displayed on the Product Name Change Dialog Box, and it is the name that will be displayed in the product tree on the left side of the ConflictSolver Main View.</p>  <p>Note • Changing this “display” name does not change the ProductName Property that appears in the title bar of the ConflictSolver Product View.</p>

Example of Importing a “Duplicate” Package

One common reason why you might want to import the same package into your Application Catalog database more than once would be if you wanted to use AdminStudio 9.5 Tuner to create custom installation SKUs of a common MSI package to distribute to different departments in your organization, each installation including certain features that are appropriate for the department and excluding certain features that are not appropriate. For example, if you were distributing a copy of Microsoft Office, you could add transforms to the Microsoft Office MSI package so that:

- Accounting's installation would include only Word and Excel
- Marketing's installation would include only Word and PowerPoint, and
- Development's installation would include only Word and Access.

Therefore, you might want to import the same package into your database more than once, each time with a different set of transformations. What happens when you import the package the second time depends upon the identifiers you selected on the Duplicate Package Tab. In this example:

- If you select the List of Transform Files and ProductCode identifiers on the Duplicate Package Tab of the Options dialog box, ConflictSolver will not identify these two packages as duplicate, even though they have the same ProductCode, because they have a different set of transformations. Therefore, the Product Name Change dialog box will not appear, and the package will be imported with the same display name as the first package.
- If you only select the ProductCode identifier on the Duplicate Package Tab of the Options dialog box, ConflictSolver will identify the second package as a duplicate because the two packages have the same ProductCode, and the Product Name Change dialog box will appear.



Note • The options that you select on the *Duplicate Package* tab of the *Options* dialog box apply globally to all packages that you attempt to import; you cannot apply different identifiers to different packages. Also, since these options are saved in the *AdminStudio Shared Directory*, everyone using *AdminStudio* at your organization who shares the same directory will share the same *Duplicate Package* options.

Extended Attributes Tab

From the *Extended Attributes* tab of the *Options* dialog box, you can specify the name and location of the extended attribute description file (.xml) which specifies the extended attributes available for each package in ConflictSolver.

AdminStudio includes a default XML file for extended attributes, which is stored in the AdminStudio shared location. You can also construct your own [Extended Attribute Description File](#). Each new Application Catalog automatically points to this file, and displays the name and location of the file in this tab.



Note • You can also configure ConflictSolver to integrate with Workflow Manager by selecting the corresponding option.



Tip • If you overwrite the default XML file with your extended attributes data, all subsequent Application Catalogs created include your attributes by default.

General Tab

From the **General** tab of the **Options** dialog box, you can configure the following options:

Table 13-90 • General Tab Properties

Properties	Description
Confirm All Drag-Drop Operations	Select this option if you want ConflictSolver to prompt you for confirmation whenever you drag and drop items.
Display Broken MSI/MST Package Links	When selected, any broken package links will be indicated by an icon change in the Product View, and with a message in the Products view, which allows you to attempt to locate the package.
Only Display View Nodes With Data	When this option is selected, packages containing views without data will not display those views. For example, if a package has no shortcuts, then the Shortcuts view is not displayed for that package.

Table 13-90 • General Tab Properties (cont.)

Properties	Description
Display RPT Reports	<p>By default, three Crystal Reports options are listed on the Reports menu and on the package context menu:</p> <ul style="list-style-type: none">• Crystal Reports Conflicts• Crystal Reports Files Report• Crystal Reports Registry <p>To suppress the display of these Crystal Reports selections—so that only the standard ConflictSolver Package, File, and Registry selections are listed—clear this option.</p>
RPT Reports: SQL Server	<p>If the Display RPT Reports option is selected, these fields are enabled to allow you to specify the location of the Crystal Reports files.</p> <p>ConflictSolver searches the directory specified for Crystal Reports reports. You can directly enter the path, or use the Browse button (...) to select it. Reports must be in Crystal Report (.rpt) format.</p>
Turn off display of conflict detection and resolution options in Application Manager	<p>By default, all functionality for interacting with the Application Catalog, including conflict detection and resolution, is available in Application Manager.</p> <p>To turn off the display of conflict detection and resolution options in Application Manager, select this option. With this option selected, all of the conflict detection and resolution functionality performed by ConflictSolver will no longer be available in the Application Manager interface. See the Comparison of ConflictSolver and Application Manager Functionality table for more information.</p>

Import Tab

From the Import tab, you can configure several options that affect how packages are imported into ConflictSolver. Most of these options can be overridden on a package-by-package basis; these options serve as the defaults.

You can configure the following options:

Table 13-91 • Import Tab Properties

Properties	Description
Check for Validation Errors	By default, packages will be checked for validation errors on import.
Check for Conflicts	If this option is selected, the Conflict Wizard will run immediately following the package import to perform conflict analysis on the newly imported package.

Table 13-91 • Import Tab Properties

Properties	Description
Automatically run PackageExpert test after import	If this option is selected, PackageExpert will test the package immediately following import to check the integrity of the package and automatically resolve any errors that are found.
Import Binary Data	Binary data in the package will be imported.
Import Custom Tables	By default, custom Windows Installer tables will be imported into ConflictSolver. Tables specified in the Ignore Tables list will not be imported.
Abort on Fatal Error	Select this option to instruct ConflictSolver to stop import if a fatal error occurs. Fatal errors include loss of network connection or running out of space. Non-fatal errors include duplicate packages or merge modules or data truncation.

Ignore Tables List

The Ignore Tables list displays all of the tables that will be ignored during import (not imported into ConflictSolver). You can select tables and delete them from the list, or you can add new tables to the list by clicking Add (which displays the Add Ignore Table dialog box).

Resolution Tab

On the Resolution Tab of the **Options** dialog box, you can specify options to determine how conflict resolutions are performed.

The **Resolution** Tab contains the following options:

Table 13-92 • Resolution Tab Properties

Option	Description
Perform resolutions against a transform file	<p>If this option is selected, ConflictSolver will, by default, save conflict detection resolutions in a transform file (MST). The original Windows Installer package (MSI) remains untouched.</p> <p>You can override this option for an individual package by going to that package's Conflict View and clicking Edit and changing this option on the Resolution Options Dialog Box.</p>

Table 13-92 • Resolution Tab Properties (cont.)

Option	Description
Preview and approve all resolutions	Select this option if you want to review and approve the suggested resolution for each conflict before ConflictSolver performs any resolutions. When the conflict detection process is completed, the Resolution Details Dialog Box is displayed, listing the conflicts that were found and the CARDS that could be used to resolve these conflicts. Before ConflictSolver will continue resolving these conflicts, each conflict resolution must be approved. See Conflict Resolution Process for more information.
Maximize user flexibility in selecting resolutions	Some CARDS can resolve a conflict in more than one way. For example, ACE06 checks whether the executable module within the component is the key file, and accumulates a list of potential files to be used for the key file. If this option is selected, this list of files is displayed on the CARD Resolution Options Dialog Box , and you are prompted to select which file to use. If this option is not selected, the CARD will perform its default resolution.



Validate Tab

From the Validate tab of the Options dialog box, you can configure the following properties:

Table 13-93 • Validate Tab Properties

Option	Description
Suppress Warnings/Info	When these boxes are checked, the messages of that type will not be displayed in the Validate tab of the Output Window during validation. It is strongly recommended you do not suppress warning messages so you can ensure proper validation of packages before importing them into ConflictSolver.

Table 13-93 • Validate Tab Properties (cont.)

Option	Description
MSI Input Files	<p>There are two options that you can configure for MSI input files:</p> <p>CUB File</p> <p>The file specified in this field is file containing Internal Consistency Evaluators (ICEs) used for validation. Either enter the location of this file directly, or use the Browse button (...) to locate it.</p> <p>ICE Rules</p> <p>By default, ConflictSolver uses all ICEs contained in the default CUB file during validation. To permanently exclude the execution of specific ICEs, list them in this edit field, separated by semicolons (;).</p> <p></p> <hr/> <p>Caution • The functionality of this field has changed. In previous versions of AdminStudio, ConflictSolver would use all of the ICEs in the default CUB file during validation unless you entered specific ICEs to run in the ICE Rules field. Now the ICE Rules field is used to exclude specific ICEs from being used during validation, meaning that all of the ICEs in the default CUB file will be used except those entered in the ICE Rules field.</p>
MSM Input Files	<p>There are two options that you can configure for merge module input files:</p> <p>CUB File</p> <p>The file specified in this field is file containing Internal Consistency Evaluators (ICEs) used for validation. Either enter the location of this file directly, or use the Browse button (...) to locate it.</p> <p>ICE Rules</p> <p>By default, ConflictSolver uses all ICEs contained in the default CUB file during validation. To permanently exclude the execution of specific ICEs, list them in this edit field, separated by semicolons (;).</p> <p></p> <hr/> <p>Caution • The functionality of this field has changed. In previous versions of AdminStudio, ConflictSolver would use all of the ICEs in the default CUB file during validation unless you entered specific ICEs to run in the ICE Rules field. Now the ICE Rules field is used to exclude specific ICEs from being used during validation, meaning that all of the ICEs in the default CUB file will be used except those entered in the ICE Rules field.</p>

History Tab

On the History tab, you can specify the type of package history information that you want to log and be able to view on a package's History View. The following options are included:

Table 13-94 • Options Dialog Box/History Tab Options

Option	Description
Maximum History Log Per Package	Enter the total number of history records you want to save for each package.
Import/Reimport	Specify whether to log and whether to display an entry each time a package is imported or reimported.
Validation	Specify whether to log and whether to display an entry each time validation is performed on a package.
Conflict Detection	Specify whether to log and whether to display an entry each time conflict detection is performed on a package.
Resolution	Specify whether to log and whether to display an entry each time a package conflict is resolved.
Extended Attribute Modification	Specify whether to log and whether to display an entry each time an extended attribute of a package is modified.
Package Description Modification	Specify whether to log and whether to display an entry each time a package description is modified.
Package Move/Copy	Specify whether to log and whether to display an entry each time a package is copied or moved from one group to another.
Patch Analysis	Specify whether to log and whether to display an entry each time patch analysis is performed on a package.

Patch Impact Information Dialog Box

When you click on one of the informational links in the **Summary** section of the [Impact Analysis View \(Patches Tab\)](#), the **Patch Impact Information** dialog box opens, displaying information on the selected link.

For example, if you click on the **Products** link, the **Patch Impact Information** dialog box opens with the **Products** tab selected.

The dialog box contains the following tabs (which are only displayed if they have data):

Table 13-95 • Patch Impact Information Dialog Box Tabs

Tab	Description
Products	The product(s) against which impact analysis was performed. The name, description, and any groups the products belong to are displayed.
Updated Products	The name, description, and modification date of any relevant product that has been updated since the last impact analysis was performed.
Deleted Products	The name and description of any relevant product that has been removed from ConflictSolver since the last impact analysis was performed.
New Products	The name, description, and import date of any product that has been added to ConflictSolver since the last Impact analysis was performed.



Note • This dialog box is similar in function to the [Conflict Information Dialog Box](#).

Resolution Details Dialog Box

The Resolution Details dialog box, which appears when you click Resolve or Resolve All on the Conflicts View, lists the proposed CARD resolutions that will be made to the package. Select the CARD resolutions that you want to execute and click Approve.

If you want to see the precise instructions that will be executed by a CARD during the resolution process before you approve it, select a proposed CARD resolution and double-click to access the Resolution Operations dialog box.



Note • The Resolution Details dialog box only appears if the Preview and approve all resolutions option is selected on the Resolution Tab of the Options dialog box. If this option is not selected, this dialog box does not appear. If you do not want this dialog box to appear, select the Don't show this dialog again check box on the dialog box. This is equivalent to unselecting the Preview and approve all resolutions option on the Resolution Tab of the Options dialog box.

Resolution Operations Dialog Box

This dialog box shows the selected CARD that is being executed, and the precise instructions that will be executed by the CARD during the resolution process.

Click Close to exit this dialog box.

Resolution Options Dialog Box

From this dialog box, you can specify the resolution method.

Table 13-96 • Resolution Options Dialog Box Properties

Properties	Description
Save resolutions in original MSI package	Select this option if you want to save conflict detection resolutions in the existing Windows Installer package.
Save resolutions in a new transform file	Select this option if you want to save conflict detection resolutions in a transform file. In the text box, specify the name and location for this transform file or click Browse and select the transform file location.

This dialog box is accessible through the Conflicts View and from the Conflict Resolution dialog box by clicking Edit in the Resolution Options section.

Rules Viewer Dialog Box

The Rules Viewer dialog box, accessible by selecting Rules Viewer from the Conflicts menu, allows you to view the current categorization of ACEs used for conflict identification. More importantly, it allows you to access the [Rules Wizard](#) to include user-defined ACEs in ConflictSolver.

The primary window in the Rules Viewer dialog box displays a tree view containing each available ACE, grouped by category. If you click New, the Rules Wizard launches, allowing you to configure information for a new ACE.

When a user-defined ACE is selected on the Rules Viewer dialog box, the Edit button is enabled. When you click Edit, the [ACE Rule Properties Dialog Box](#) appears, where you can reconfigure the ACE. You can also delete user-defined ACEs by selecting them and clicking Delete.

Wizards

The following Wizards are included in ConflictSolver:

- [Conflict Wizard](#)
- [Rules Wizard](#)
- [Validation Wizard](#)

Conflict Wizard

Although a Windows Installer package or merge module may be built to guidelines put forth by Microsoft, it is possible that the interaction between packages, or between a package and the base operating system, may cause unwanted results in your production environment. You can use the Conflict Wizard to identify these conflicts before you deploy packages, and resolve the problems before they affect your end users.

The Conflict Wizard allows you to identify conflicts between a Windows Installer package and packages already cataloged in ConflictSolver. You can check for a variety of conflict types, including file, component, and registry conflicts. In many cases, ConflictSolver can resolve the issues automatically. You can also create your own custom rules to ensure packages conform to your internal standards and practices. ConflictSolver has rules to detect conflicts involving: Components, Files, Registry Entries, Shortcuts, INI Files, ODBC Resources, NT Services, File Extensions, and Product Properties.

The Conflict Wizard consists of the following panels:

- [Welcome Panel](#)
- [Source Type Panel](#)
- [MSI Source Information Panel](#)
- [Source Package Panel](#)
- [Choose Action Panel](#)
- [Best Practice Rules Panel](#)
- [Conflict Rules Panel](#)
- [Target Information Panel](#)
- [Summary Panel](#)

When run, ConflictSolver displays the output report in the Conflicts tab of the Output Window.

Welcome Panel

The first panel of the Conflict Wizard welcomes you to the Wizard.

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:

Table 13-97 • Welcome Panel Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific Conflict Wizard panel.




Source Type Panel

From the Source Type panel, select whether you want to compare an external Windows Installer package against packages already in the ConflictSolver Application Catalog or perform a comparison between packages within the ConflictSolver Application Catalog.

- If you select Internal Application Catalog package, the next panel that appears is the Source Package panel.
- If you select Windows Installer package, the next panel that appears is the MSI Source Information Panel.

MSI Source Information Panel

From the MSI Source Information Panel, specify the Windows Installer package you want to check for conflicts against packages already in ConflictSolver.

Click Browse to locate and select the package. Should that package require transforms, click the New button () above the Transforms area. Use the Browse button (...) to locate the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up () and Move Down () buttons.

Delete a transform file from the Transforms list by selecting the transform and clicking the Delete button ()

Source Package Panel

When you select the Internal Application Catalog package option on the Conflict Wizard Source Type Panel (specifying that you want to compare source packages from within the ConflictSolver database), the Source Package panel appears.

This panel displays the packages in the same hierarchy as the Product View. For your convenience, empty groups are not displayed.

The ConflictSolver supports multiple source package checking. Therefore, select the package(s) you want to analyze for conflicts. If you select multiple source packages, ConflictSolver will conflict check each source package serially against all of the selected target packages.



Note • The target packages are specified on the next panel, the Target Information Panel. The Target Information panel will naturally exclude all packages that you selected on the Source Package panel.

You can also select all packages in ConflictSolver or clear all selected packages using the Select All and Clear All buttons.



Caution • Since the Target Information panel excludes all packages that are selected on the Source Package panel, exercise caution when using the Select All button. If you select all of the packages on the Source Package panel, there will be no target packages listed on the Target Information Panel, and therefore you will not be able to run conflict detection.

Choose Action Panel

On this Panel you specify whether to evaluate the selected source packages against Best Practice rules or against other packages. You can also specify whether you want ConflictSolver to automatically resolve resolvable conflicts, and whether to automatically generate a report when conflict analysis is complete.

The following options are included:

Table 13-98 • Choose Action Panel Options

Options	Description
Evaluate source package against best practice rules	Select this option to include Best Practice ACE rules in this conflict analysis.
Detect conflicts between source package and other packages	Select this option to include Conflict ACE rules in this conflict analysis.
Run Resolution	Select to display the Conflict Resolution dialog box after conflict detection. From the Conflict Resolution dialog box, you can access information pertaining to the last execution of the Conflict Wizard. This persisted conflict data allows you to view when the last execution was performed, the products that it was run against, the ACE rules used, and conflicts discovered. The dialog box also provides a list of updated, deleted, or added products that may necessitate performing conflict identification again.
Generate Reports	Select this option to generate a conflict detection analysis report containing a summary of the conflicts that were detected, and a detailed description of each identified conflict including how this conflict will be resolved.

Best Practice Rules Panel

ConflictSolver includes Best Practice rules that operate like ICEs (Internal Consistency Evaluators) to enforce Microsoft best practices. While an ICE is a custom action written by Microsoft which is used during validation to determine if an installation package is built according to Windows Installer standards, an ACE is a custom action included with ConflictSolver that is also used to enforce Windows Installer standards.

In the Best Practice Rules Panel, select the types of Best Practice Rules to evaluate in conflict analysis. By default, the Best Practice ACEs that are selected on the [Rules Tab](#) of the Options dialog box will be selected.

The following Best Practice ACE rules are available.

Table 13-99 • Best Practice ACE Rules

Conflict Type	Associated Rules
Components	ACE04 , ACE05 , ACE06 ,

Table 13-99 • Best Practice ACE Rules

Conflict Type	Associated Rules
Merge Module Integrity	ACE26 , ACE36
Recommended Rules	ACE25 , ACE27 , ACE28 , ACE29 , ACE31 , ACE32 , ACE33 , ACE34 , ACE35

Conflict Rules Panel

In the Conflict Rules panel, select the types of conflicts to use for conflict evaluation. The following Conflict ACE rules are available:

Table 13-100 • Conflict ACE Rules

Conflict Type	Associated Rules
Component	ACE02 , ACE04 , ACE05 , ACE06 , ACE09
Files	ACE03 , ACE07 , ACE08 , ACE12 , ACE23
Registry	ACE10 , ACE24
Shortcuts	ACE13
INI Files	ACE14 , ACE21 , ACE22
ODBC Resources	ACE15
NT Services	ACE16
File Extensions	ACE17
Product Codes	ACE18 , ACE19 , ACE20



Note • The default settings in the Conflict Options panel reflect the settings in the Conflicts tab of the Options dialog box (available from the Tools menu).

Target Information Panel

In the Target Information Panel, select the individual packages or groups of packages in ConflictSolver that you want to compare the source package(s) against.

Each package selected will be compared against the packages you specified in the Source Package Panel (for internal comparisons) or MSI Source Information Panel (for comparisons with an external Windows Installer package).



Note • The Target Information panel excludes all packages that you selected on the Source Package panel. Empty groups are also excluded.

You can also select all packages in ConflictSolver or clear all selected packages using the Select All and Clear All buttons.

Summary Panel

The Summary panel provides a detailed summary of the options that were selected in the previous panels of the Wizard.

Click Finish to run the Conflict Wizard using the options specified.

Rules Wizard

The Rules Wizard allows you to create user-defined ACE rules for later use by the ConflictSolver Wizard. It is accessible by clicking New on the Rules Viewer dialog box.

The following panels are part of the Rules Wizard:

- [Welcome Panel](#)
- [General Information Panel](#)
- [Additional Information](#)
- [Custom Options Panel](#)
- [Where Clause Panel](#)
- [DLL-Based ACEs Panel](#)
- [Summary Panel](#)

Welcome Panel

The Rules Wizard allows you to create user-defined ACE rules for later use in conflict identification. The first panel displayed is the Welcome panel.

Click Next to proceed to the General Information Panel.

General Information Panel

From the General Information panel, you can configure information about the new ACE rule. This information is used primarily for display information (Name, Brief Description, Description, and Information URL).

Table 13-101 • General Information Panel Options

Option	Description
Name	The name of the ACE, used to organize the rule in ConflictSolver. This is displayed in several places, including the Output Window, the Rules Viewer dialog box, and the Conflicts tab of the Options dialog box.
Associated Table	Select the table in the Application Catalog which will be queried in the user-defined ACE. This also determines which columns are available in the Expression Builder dialog box, and which tokens are available for the Error and Display strings on the Custom Options Panel of the Rules Wizard.
Brief Description	Enter a brief description which will be displayed in the Rules Viewer, the Conflicts tab of the Options dialog box, and in the Output Window. This description should be clear enough so users can understand when to use this ACE.
Description	Enter a description of the ACE, which is displayed at the bottom of the Rules Viewer dialog box when the ACE is selected and in the Output Window during conflict identification when the ACE executes.
Information URL	Provide a URL to get further information for the ACE. This URL appears in the Conflict Details area of the Conflicts View after conflicts have been identified.

Click Next to proceed to the Additional Information Panel; click Back to return to the Welcome Panel.

Additional Information

From the Additional Information Panel, you can provide information for categorizing the ACE in relation to other ACEs in ConflictSolver.

Table 13-102 • Additional Information Panel Option

Option	Description
Category	Either select an existing category for this new rule, or enter the name for a new category. These categories are displayed in the Conflict View and the Conflicts tab of the Options dialog box. Ideally, any user-defined ACEs should be put in their own category.

Table 13-102 • Additional Information Panel Option (cont.)



Option	Description
Rule Type	<p>Specify the type of ACE you are creating:</p> <ul style="list-style-type: none"> Custom - Source Only Packages ACEs allow you to quickly test any column or any value of a table to support your business logic. For example, you could use a user-defined ACE to identify packages that create a desktop icon. To define a Source Only Packages ACE, you must define an SQL “Where” clause. <p>ConflictSolver supports external package conflict checking for Custom - Source Only Packages ACEs. The Source package can be selected from the ConflictSolver Application Catalog Database or from an external MSI package.</p> <p>See Creating a Custom/Source Only Packages ACE for more information.</p> Custom - Source and Target Packages ACEs allow you to compare columns or values of Source package tables (new packages that you want to install onto a user’s system) to columns or values of Target package tables (packages already installed on a user’s system). <p>For example, you could use a Source and Target Packages ACE to determine if the installation of a Source package onto a Target system would overwrite or conflict with an existing entry in the .ini file in the System directory of the Target system.</p> <p>To define a Source and Target Packages ACE, you must define a SQL “Where” clause, and specify a Join Column—a table column in the Application Catalog database that has a matching value for both the Source and Target packages. Rows in each of the packages that have a matching value in the Join Column are selected and those rows are checked against the Source and Target Packages.</p> <p>ConflictSolver <i>does not support</i> external package conflict checking for Custom - Source and Target Packages ACEs. Both the Source and Target Packages must be selected from the ConflictSolver Application Catalog Database.</p> <p>See Creating a Custom/Source and Target Packages ACE for more information.</p> DLL - User Provided DLL Based ACEs allow you to run more complex tests—testing many tables in any combination. For example, you could use a DLL-Based ACE to confirm that a source product language is the same as all target product languages. To define a DLL-Based ACE, you use SQL and various programming languages to construct a Windows DLL. With DLL-Based ACEs, you can use a Conflict Application Resolution Definitions (CARs) to fix the conflict. See Creating a User Provided DLL-Based ACE.

Click Next to proceed to the Custom Options Panel or the DLL-Based ACEs Panel; click Back to return to the General Information Panel.

Custom Options Panel

From the Custom Options panel, you can create display strings for the Output Window and Conflict Details.

Table 13-103 • Custom Options Panel Option

Option	Description
Error String	<p>This string appears in the Output Window when a violation of this ACE rule is detected during conflict identification. For example, if you were creating a user-defined ACE to identify packages that create a desktop icon, you could enter Failure in creating desktop icon in this field.</p>  <p>Note • Tokens allow you to insert values at run-time from the internal ConflictSolver Application Catalog Database or an external MSI package into the string, such as specifying a file name. To use token replacement in the error string, click on the arrow to the right of the Error String field and pick a value from the list, or just type the values directly in the text box, in the following format:</p> <ul style="list-style-type: none"> • Source Only Packages ACEs: [ColumnName] • Source and Target Packages ACEs: [Source.ColumnName] and [Target.ColumnName] <p>For more information, see Token Grammar.</p>
Display String	<p>This string appears in the Conflict Details area of the Conflicts View after conflicts have been identified. For example, if you were creating a user-defined ACE to identify packages that create a desktop icon, you could enter Duplicate desktop icon found in this field.</p>  <p>Note • Tokens allow you to insert values at run-time from the internal ConflictSolver Application Catalog Database or an external MSI package into the string, such as specifying a file name. To use token replacement in the error string, click on the arrow to the right of the Display String field and pick a value from the list, or just type the values directly in the text box, in the following format:</p> <ul style="list-style-type: none"> • Source Only Packages ACEs: [ColumnName] • Source and Target Packages ACEs: [Source.ColumnName] and [Target.ColumnName] <p>For more information, see Token Grammar.</p>
Severity	Specify whether this ACE should be an Error or a Warning.
Report 'No' results	User defined ACEs report conflicts based on the provided query. However, you may want to report the absence of the data if it could not be found. If you select this option, if the ACE does not return any results, it will be reported as an error (or warning), with the description and error strings as specified. If you expect a No result, do not use tokens in your display or error strings.

Click Next to proceed to the Where Clause Panel; click Back to return to the Additional Information Panel.

Token Grammar

What are Tokens?

Tokens represent data in the database that is inserted at runtime. In ConflictSolver, Tokens are used to insert values at runtime from the ConflictSolver Application Catalog Database or an external MSI package into an Error or Display String.

How to Insert Tokens

Tokens are specified on the Custom Options panel of the Rules Wizard. To use token replacement in a string, click the arrow to the right of the Error String and Display String text boxes and select a column name from the list. The column name is then inserted into the string in the following format:

- **Source Only Packages ACEs**—[ColumnName]
- **Source and Target Packages ACEs**—[Source.ColumnName] and [Target.ColumnName], with the prefix identifying whether the column is in the Source or Target package. If no prefix is used, ConflictSolver assumes the “Source.” prefix.



Note • The Token list on the Custom Options panel is provided for your convenience; if you prefer, you can type the variables directly in the text boxes. For more information see, [Token Grammar](#).



Caution • While you are creating a user-defined ACE in the Rules Wizard, if you initially select a Rule Type of Custom - Source and Target Packages, and then insert tokens in the Error String and Display String fields, the “Source.” prefix will be used. But, before you finish creating this ACE, if you go back and change your Rule Type selection to Custom - Source Only Packages, the tokens that you initially entered into the Display and Error string text boxes will not automatically be updated to remove the “Source.” prefix. For ConflictSolver to correctly interpret this ACE, you need to manually go back to the Error and Display String fields and delete the “Source.” prefix.

Using the ProductName Pseudo-tokens

You can use the pseudo-tokens of [ProductName], [Source.ProductName] and [Target.ProductName] to insert the name of the Source or Target package in an Error or Display String, even though ProductName is not a table column name.

Where Clause Panel

From the Where Clause Panel, you must define a valid Where clause for the ACE. If you do not know how to build a Where clause, you can click the Build Expressions button to launch the Expression Builder dialog box. You can also click Test to validate the Where clause syntax.

If you selected **Custom - Source and Target Packages** on the Additional Information Panel, you must also select a **Join Column**—a table column in the Application Catalog database that has a matching value for both the Source and Target packages. Rows in each of the packages that have a matching value in the Join Column are

selected and those rows are checked against the Source and Target Packages. For example, if you wanted to evaluate Source and Target packages that write files to the same directory, you might specify Directory as the Join column.



Tip • To improve query performance, enclose table names in square brackets ([]).

DLL-Based ACEs Panel

DLL-based ACEs require specific information about the ACE/CARD DLL file and the ACE and CARD Function Names to operate. Enter the following information:

Table 13-104 • DLL-Based ACEs Panel Options

Option	Description
ACE/CARD DLL File	Select the name of the ACE DLL that you are testing.
ACE Function Name	Enter the name that you chose to “export” for this ACE function.
CARD Function Name	Enter the name that you chose to “export” for this CARD function.
Test	Click the Test button next to the ACE Function Name or CARD Function Name to validate that the exported function does exist.

Click Next proceed to the Summary Panel; click Back to return to the Additional Information Panel.

Summary Panel

Once you have configured information for your ACE rule, the Summary panel displays information for final review.

Click Finish to accept this configuration and make the ACE available for conflict identification. Click Back to return to either the **Where Clause Panel** or the DLL-Based ACEs Panel

Validation Wizard




The Validation Wizard allows you to validate a Windows Installer package and any associated transforms prior to importing them into ConflictSolver. Validation is performed against the validation file (.cub) selected in the Validate tab of the Options dialog (available from the Tools menu).


The following panels are included in the Validation Wizard:

- [MSI Source Information Panel](#)
- [Summary Panel](#)

MSI Source Information Panel

From the MSI Source Information Panel, specify the Windows Installer package you want to check for conflicts against packages already in ConflictSolver.

Click Browse to locate and select the package. Should that package require transforms, click the New button () above the Transforms area. Use the Browse button (...) to locate the transform. If the package requires multiple transforms, you can repeat the procedure as necessary. The order in which transforms are applied can be changed by selecting a transform and clicking the Move Up () and Move Down () buttons.

You can delete a transform from the Transform files list by selecting the transform and clicking the Delete button ()

Click Next to proceed to the Summary Panel.

Summary Panel

From the Summary panel, you can review the Windows Installer package and any associated transforms you will validate against the specified CUB file.

If the options are correct, click Finish to begin validation. If not, click Back to return to the MSI Source Information Panel.

Identifying and Resolving Package Errors Using PackageExpert



Edition • PackageExpert is included with AdminStudio Enterprise Edition.

PackageExpert is the latest addition to AdminStudio's array of powerful testing tools. You can use PackageExpert to check the integrity of your Windows Installer packages and automatically resolve errors that are found.

PackageExpert allows you to quickly run and resolve a series of tests on Windows Installer packages to improve their overall quality and their deployment readiness. PackageExpert provides the flexibility to augment and extend the test suite to suit any particular end user requirement. PackageExpert's tests are currently organized into three major categories:

- **ICE Tests**—The ICE tests perform checks against the structure of a Windows Installer package to enforce Microsoft best practices. They are used to determine if an installation package is built according to Windows Installer standards.
- **Windows Vista Certification Tests**—The Windows Vista tests check packages for compliance with installation requirements of the Windows Vista operating system, and are intended to enable the successful deployment of packages on Vista.
- **Windows 7 Compatibility Tests**—The Windows 7 Compatibility tests check packages for compliance with installation requirements of the Windows 7 operating system, and are intended to enable the successful deployment of packages on Windows 7.

PackageExpert documentation is organized in the following sections:

Table 14-1 • PackageExpert Help Library

Section	Description
About PackageExpert	Introduces you to PackageExpert, its interface, and how it works.
Configuring PackageExpert	Explains how to specify which tests to run by default every time a package is tested.

Table 14-1 • PackageExpert Help Library

Section	Description
Testing Packages and Viewing Results	Explains how to test both standalone packages and packages in an Application Catalog. Explains how to view test results in PackageExpert and in ConflictSolver. It also explains how to customize your test results by excluding specific errors and by changing the severity of specific errors.
Resolving Errors	Explains how PackageExpert automatically resolves errors, and how you specify how you want errors to be resolved.
Adding Manual Tests	Provides instructions on how to add and run your own XML-based tests to PackageExpert.
PackageExpert Reference	Every view and dialog box available in PackageExpert is discussed in this section.

About PackageExpert

PackageExpert allows you to quickly run and resolve a series of tests on Windows Installer packages to improve their overall quality and their deployment readiness. PackageExpert provides the flexibility to augment and extend the test suite to suit any particular end user requirement. PackageExpert's tests are currently organized into three major categories:

- **ICE Tests**—The ICE tests perform checks against the structure of a Windows Installer package to enforce Microsoft best practices. They are used to determine if an installation package is built according to Windows Installer standards.
- **Windows Vista Certification Tests**—The Windows Vista tests check packages for compliance with installation requirements of the Windows Vista operating system, and are intended to enable the successful deployment of packages on Vista.
- **Windows 7 Compatibility Tests**—The Windows 7 Compatibility tests check packages for compliance with installation requirements of the Windows 7 operating system, and are intended to enable the successful deployment of packages on Windows 7.

In addition, you can add your own XML-based manual tests.

The following diagram provides an overview of how you can use PackageExpert to perform testing and resolution:

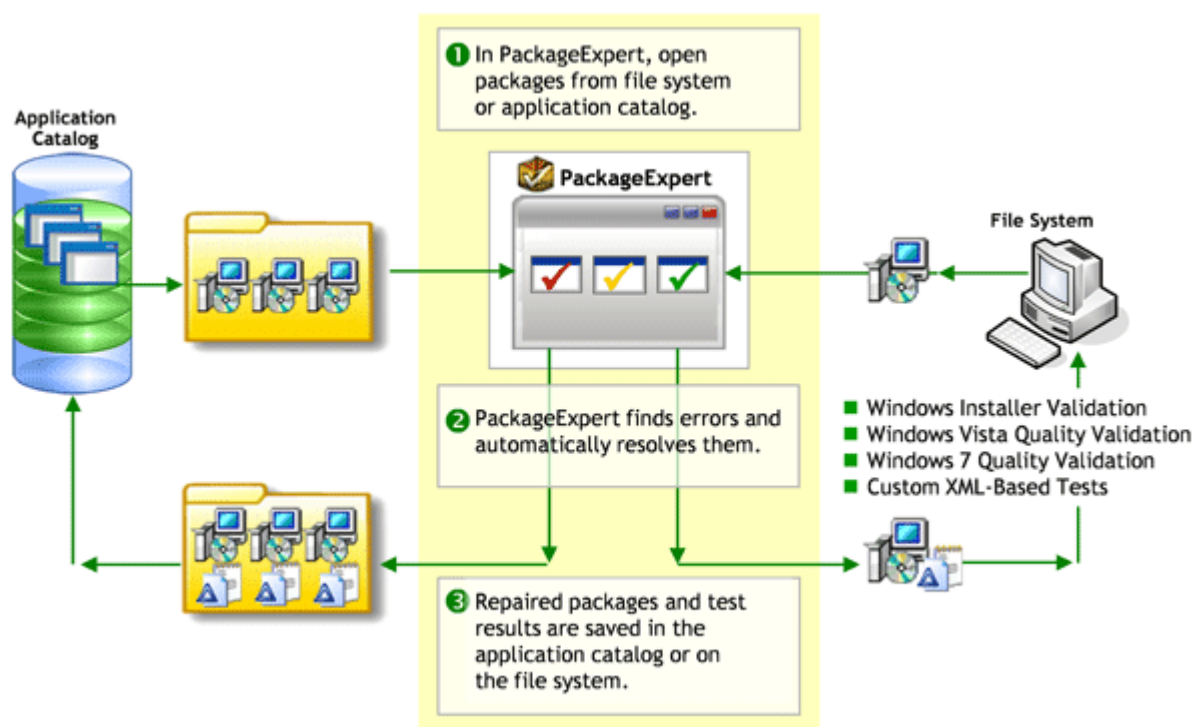


Figure 14-1: PackageExpert Overview Diagram

Using PackageExpert to perform testing is a three-step process:

- **Step 1: Select Package(s)**—Open a package from the file system or connect to an Application Catalog database.
- **Step 2: Run Tests**—Select a package or group folder and select **Test Package** from the context menu.
- **Step 3: Resolve Errors**—PackageExpert can automatically resolve some errors, while others require you to perform a manual resolution:
 - **To resolve those errors that are automatically resolvable**, select a test category, package, or message node and select **Resolve All** or **Resolve Test** from the context menu.
 - **To resolve errors requiring manual resolution**, select the message node and perform the instructions provided in the right pane under **Resolution: Manual**.

Configuring PackageExpert

You can specify which tests you want PackageExpert to run for subsequent test runs, and you can also specify whether to resolve errors directly in the original Windows Installer .msi file or using a transform .mst file.

- [Setting Default Test Configuration](#)
- [Specifying the Resolution Method](#)

Setting Default Test Configuration

On the **Configuration** view, you specify which tests you want PackageExpert to run for all subsequent test runs. These settings can be changed at any time, but any changes you make do not effect the results of packages that have already been tested.

The **Configuration** view consists of two panes:

- **Package tree**—Tests are grouped into categories and are displayed in a tree on the left pane.
- **Help pane**—When you select a test, a help topic explaining that test opens in the right pane.

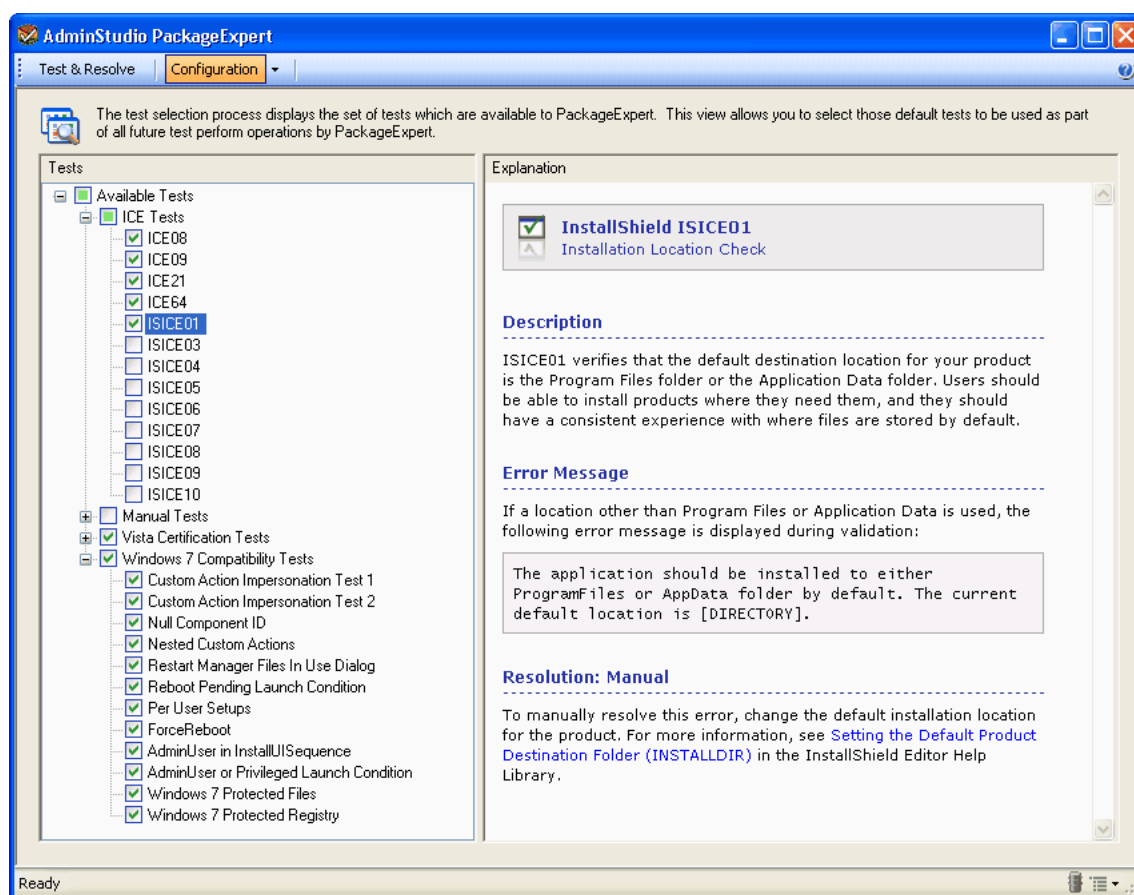


Figure 14-2: PackageExpert Configuration View

To set the default tests that will be run, perform the following steps.

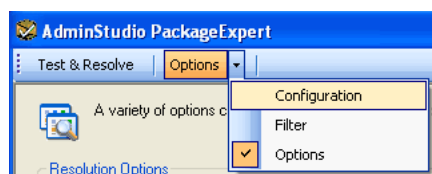


Task: *To set default tests to be run:*

1. Click **Configuration** in the PackageExpert toolbar. The **Configuration** view opens.



Note • The second menu item in the PackageExpert toolbar toggles between **Configuration**, **Options**, and **Filter**. If **Configuration** is not displayed, click the arrow next to the menu item and select **Configuration** from the list.



2. Select the tests that you want PackageExpert to run for subsequent test runs.



Tip • You can select or unselect all of the tests in a test category by selecting or unselecting the test category check box.

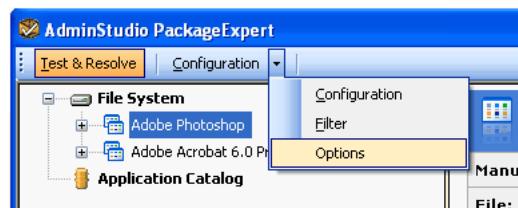
Specifying the Resolution Method

On the **Options** view, you can specify whether you want to resolve errors by creating a transform (.mst) file or by directly editing the .msi package.



Task: *To specify the resolution method:*

1. Open PackageExpert.
2. In the toolbar, click the down arrow next to **Configuration** and select **Options**. The **Options** view opens.



3. Select one of the following options to specify the resolution method you want PackageExpert to use when automatically resolving errors:
 - **Original MSI file**—Select this option to have PackageExpert directly edit the .msi file to resolve errors.
 - **A newly created transform (MST) file**—Select this option to have PackageExpert create a transform (.mst) file to resolve errors.

Testing Packages and Viewing Results

When testing with PackageExpert, you can choose to test one external Windows Installer package, or a selected group of packages in an Application Catalog. Also you can view test results in both PackageExpert and ConflictSolver, and can customize those results by excluding errors or changing the severity of errors.

Table 14-2 • Testing Packages and Viewing Results

Section	Description
Testing Packages	<p>This section includes the following topics:</p> <ul style="list-style-type: none"> • Testing External Packages • Testing Packages in an Application Catalog • Testing Packages From ConflictSolver
Viewing and Managing Test Results	<p>This section includes the following topics:</p> <ul style="list-style-type: none"> • Viewing Test Results in PackageExpert • Viewing Test Results in ConflictSolver • Publishing Test Results to the Application Catalog • Deleting Test Results • Customizing Test Results • Adding Ad-Hoc Test Results

Testing Packages

You can use PackageExpert to test packages that are stored on a local file system or in an Application Catalog. You can also select a package in ConflictSolver and use PackageExpert to test it.

- [Testing External Packages](#)
- [Testing Packages in an Application Catalog](#)
- [Testing Packages From ConflictSolver](#)

Testing External Packages

You can test an external package using PackageExpert.

- **Multiple packages can be open**—You can have multiple packages open PackageExpert at the same time. All will be listed in the **File System** package tree.

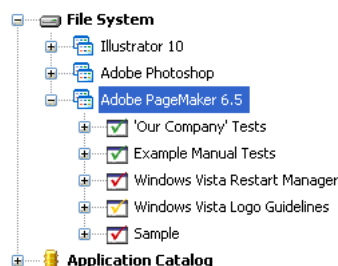


Figure 14-3: Package Tree With Multiple External Packages Open

- **Packages will remain open across sessions**—Each time you open PackageExpert, the packages that were open when you exited previously will remain open.
- **You can remove packages**—You can remove a package from the **File System** package tree by selecting it and then selecting **Remove Package** from the context menu. This action does not affect the package's underlying Windows Installer file; it is not deleted.

To test a package that is stored on a local file system, perform the following steps.



Task: *To test an external package:*

1. Select the **File System** node in the **Test & Resolve** view and select **Open Package** from the context menu. The **Select a Package** dialog box opens.
2. Click **Browse** and select the Windows Installer package that you want to test.
3. Optionally, select transform files (.mst) to include in the test process.

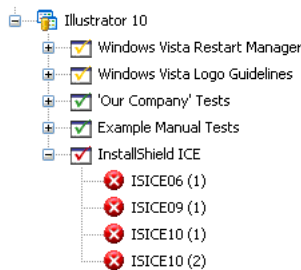
If any transform files exist in the same directory as the .msi file you selected, they are automatically listed in the **Transform List**, but are not selected.

- **To include listed transform files**—Select the transform files that you want to include in PackageExpert testing.
 - **To add additional transform files**—Click **Browse** and select a transform file to add it to the Transform list, and then select it to include it in testing.
4. Click **Test** to open the selected package and perform testing on the package.



Note • After you have selected a package, the **Test** button becomes enabled. If you click **Test**, the tests will be run. However, if you click **Cancel**, the package will open, but the Test Category nodes will not be displayed below the package icon, and the right pane will state that none of the tests have been run.

Messages are listed in the Output window, and for each error that is found, a message node is listed under the appropriate Test Category node:



If one test finds more than one issue in the package that produces an error, multiple message nodes, sequentially numbered, are listed under the **Test Category** node.

To view specific information about an error message, select it and view the **Error Details** in the right pane:

Error Details	
Item	Details
Test Name	Restart Manager Files In Use Dialog / Error Number 1
Description	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista.
Status	Ran
Details	Checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.
Help	TestHelpTopics\Restart Manager Files In Use Dialog.htm

For more information on test results, see [Viewing and Managing Test Results](#).

Testing Packages in an Application Catalog

To open an Application Catalog and test a package or group of packages, perform the following steps:



Task: *To test packages in an Application Catalog:*

1. On the **Test & Resolve** view, select the **Application Catalog** node. The **Connect Application Catalog** dialog box opens and you are prompted to log on to either the AdminStudio Enterprise Server database or a standalone Application Catalog database.
2. Connect to an Application Catalog as described in [Connecting to an Existing Application Catalog](#). When you are connected, that Application Catalog's groups and packages are listed in the package tree under the **Application Catalog** node.

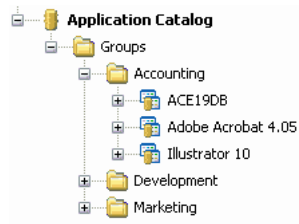
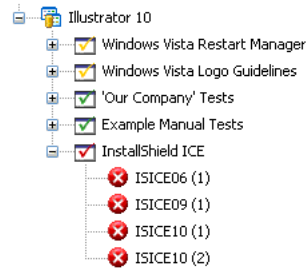


Figure 14-4: Package Tree With an Application Catalog Open

3. To test a package or group of packages in the Application Catalog, select either a package or a package group and select **Test Package** or **Test Group** from the context menu

Messages are listed in the Output window, and for each error that is found, a message node is listed under the appropriate Test Category node:



If one test finds more than one file or component in the package that produces an error, multiple message nodes, sequentially numbered, are listed under the **Test Category** node.

To view specific information about an error message, select it and view the **Error Details** in the right pane:

Error Details	
Item	Details
Test Name	Restart Manager Files In Use Dialog / Error Number 1
Description	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista.
Status	Ran
Details	Checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.
Help	TestHelpTopics\Restart Manager Files In Use Dialog.htm

For more information on test results, see [Viewing and Managing Test Results](#).

Testing Packages From ConflictSolver

You can select a package in ConflictSolver and then choose to test the package using PackageExpert. Test results will be displayed in the ConflictSolver **Test Results** view.

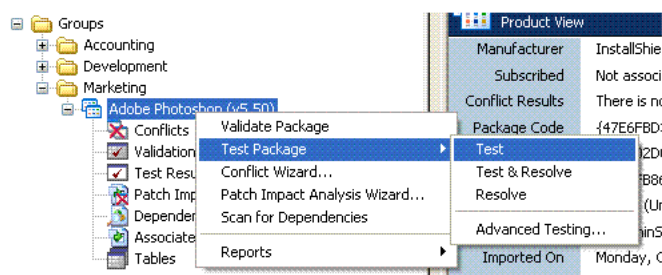


Note • By default, all functionality for interacting with the Application Catalog, including conflict detection and resolution, is available in Application Manager. Unless the **Turn off display of conflict detection and resolution options in Application Manager** option on the **General** tab of the **Options** dialog box is selected, you can also perform these PackageExpert tasks using the Application Manager interface. See [Application Manager and ConflictSolver](#) for more information.



Task: To test a package and resolve errors from ConflictSolver:

1. Open the ConflictSolver **Product** view.
2. Select the package that you want to test, point to **Test**, and select one of the following options from the context menu:

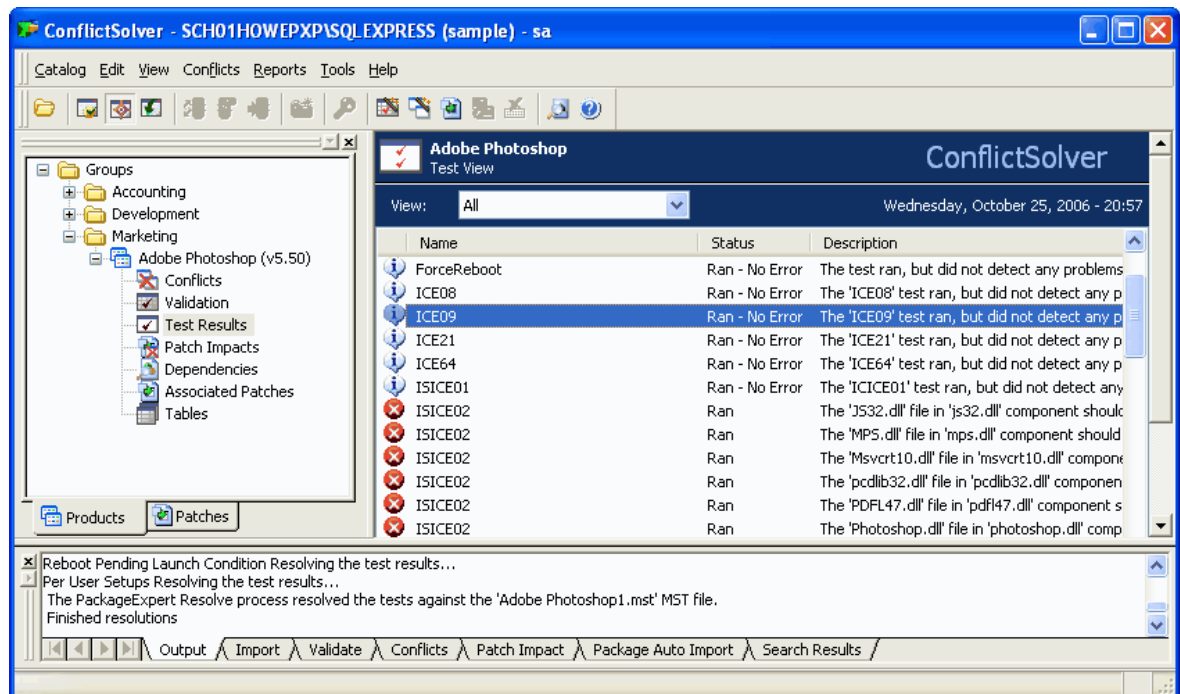


- **Test**—Select to have PackageExpert test the package but not resolve the automatically resolvable errors.
- **Test & Resolve**—Select to have PackageExpert test the package and resolve the automatically resolvable errors.
- **Resolve**—Select to have PackageExpert resolve the package's automatically resolvable errors.
- **Advanced Testing**—Select to open this package in PackageExpert and perform the testing using that interface. You will be prompted to connect to the database that you are currently connected to in ConflictSolver

If you select **Test** or **Test & Resolve**, the following message will appear in the ConflictSolver Output window, followed by other progress messages.

PackageExpert is testing the 'MyPackageName' package...

3. When testing is complete, select the **Test Result** node in the package tree to open the ConflictSolver **Test Results** view.



This list of messages can be filtered by **All**, **Ran - No Error**, **Ran**, or **Resolved** by making a selection from the **View** list.

Viewing and Managing Test Results

You can view test results in both PackageExpert and ConflictSolver. In PackageExpert, you can change the severity of a message, exclude messages from the listing, and add ad-hoc test results. For packages in an Application Catalog, you can publish a package's test results in the catalog.

- [Viewing Test Results in PackageExpert](#)
- [Viewing Test Results in ConflictSolver](#)
- [Publishing Test Results to the Application Catalog](#)
- [Deleting Test Results](#)
- [Customizing Test Results](#)
- [Adding Ad-Hoc Test Results](#)

Viewing Test Results in PackageExpert

You view both individual and summary test results in the **Test & Resolve** view. The information displayed varies depending upon what is selected in the package tree:

- [Information Shown for an Error Message](#)
- [Test Result Views](#)
- [Toggling Between Local and Published Test Results](#)

Information Shown for an Error Message

Test result messages are displayed in the right pane of the **Test & Resolve** view when a **Package**, **Test Category**, or **Message** node is selected. For each error that is found, a message node is listed under the appropriate Test Category node:

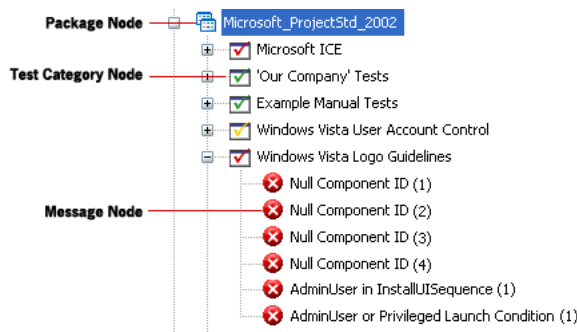


Figure 14-5: Nodes in the PackageExpert Test & Resolve View

If one test finds more than one file or component in the package that produces an error, multiple message nodes, sequentially numbered, are listed under the test category node. In the above example, the **Null Component ID** test in the **Windows Vista Logo Guidelines** test category produced four errors.

Information Shown in Summary Listings


When a **Package** or **Test Category** node is selected, a summary listing of all of the errors found in that package or test category is shown. For each error, the following information is displayed, explaining the cause of the error:

	Name	Status	Description	Help
✖	ICE08	Ran	The 'Global_Authoring_OPCRemoveMax' component has the same ComponentId as another component and it probably shouldn't.	Info

Figure 14-6: Message Information in Summary Listing

If you click the **Info** link in the **Help** column, a help topic explaining the details of the test and how it is resolved opens.

If a test has not yet been run, a message for the test appears in the list with a **Status** of **Not Run**

	Name	Status	Description	Help
	Desktop Icons	Not run	The test has not yet been run.	Info

Information Shown When Message Node is Selected

When a message node is selected, **Error Details** for that error message are displayed in the right pane. Additional details about the test are provided, as is a link to a help topic explaining that test.

Error Details	
Item	Details
Test Name	Restart Manager Files In Use Dialog / Error Number 1
Description	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista.
Status	Ran
Details	Checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.
Help	TestHelpTopics\Restart Manager Files In Use Dialog.htm

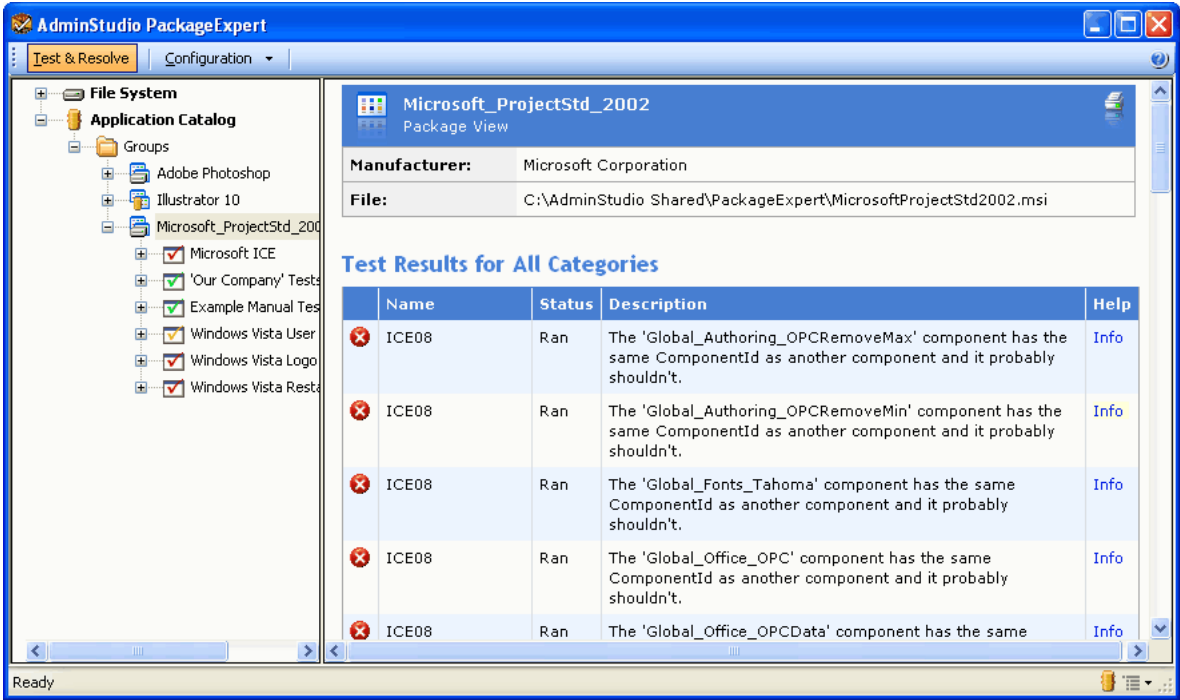
Test Result Views

The test result information displayed varies depending upon what is selected in the package tree, varying between summary information and detailed information on one error message.

- [Package View](#)
- [Group View](#)
- [Application Catalog View](#)
- [Test Category View](#)
- [Message/Error View](#)

Package View

When a package is selected in the package tree, the test results for all Test Categories are displayed, and are listed in the order of severity:



Group View

When a group in an Application Catalog is selected, a graphic illustrating the number of errors and warnings generated during testing for each package is listed:

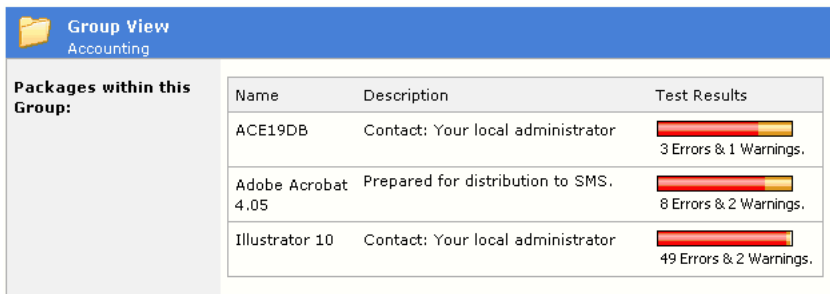


Figure 14-7: Group Summary Information

Application Catalog View

When the Application Catalog node is selected, and you are connected to an Application Catalog, two graphics illustrating the following statistics appear:

- **Percentage of Application Catalog Tested**—The percentage of the number of packages in the Application Catalog that have been tested is listed.
- **Number of Errors and Warnings**—The total number of error and warning messages generated during the testing of packages in the Application Catalog is listed.

Application Catalog View ABC04SMITH\XP\SQL\EXPRESS-sample		
Statistics:	The Application Catalog contains: <ul style="list-style-type: none"> • 8 MSI packages. 	
Most Recently Tested Packages:	Details	Date
	ACE19DB	10/24/2006 7:27 PM
	Illustrator 10	10/24/2006 7:27 PM
	Adobe Acrobat 4.05	10/24/2006 6:00 PM
Percentage of Application Catalog Tested:	<div> <div style="width: 100%;"></div> 100% (8 out of 8 MSI Packages Tested). </div> <div> <div style="width: 66.66%;"></div> 6 package(s) with Errors & 6 package(s) with Warnings. </div> <div> <div style="width: 10px; height: 10px; background-color: red; border: 1px solid black;"></div> Error </div> <div> <div style="width: 10px; height: 10px; background-color: orange; border: 1px solid black;"></div> Warning </div>	
Application Catalog History:	Details	Date
	Database created.	9/28/2006 7:37 AM

Figure 14-8: Application Catalog Summary Information

Test Category View

When a Test Category node is selected test results for all of the tests in that category are listed, and are listed in order of severity.

Test Results for Category: Windows Vista Logo Guidelines









	Name	Status	Description	Help
	Null Component ID	Ran	Component Global_Access_Core has a null ComponentID	Info
	Null Component ID	Ran	Component Global_FrontPageCore_CoreClient has a null ComponentID	Info
	AdminUser in InstallUISequence	Ran	The installation has actions in the InstallUISequence table that have conditions using the 'AdminUser' property.	Info
	AdminUser or Privileged Launch Condition	Ran	The installation has LaunchCondition entries using the 'AdminUser' or 'Privileged' properties.	Info
	Nested Custom Actions	Ran - No Error	The 'Nested Custom Actions' test ran, but did not detect any problems.	Info
	Per User Setups	Ran - No Error	The test ran, but did not detect any problems.	Info
	ForceReboot	Ran - No Error	The test ran, but did not detect any problems.	Info

Figure 14-9: Test Category Summary Information

Message/Error View


When a message node in the package tree is selected, detailed information on that specific message is displayed in the top half of the pane—including the exact text of the selected error message, and a help topic that explains what the test is checking for and how errors are resolved is displayed in the lower half of the pane.


Microsoft_ProjectStd_2002
Restart Manager Files In Use Dialog

Manufacturer: Microsoft Corporation
File: C:\AdminStudio\Shared\PackageExpert\MicrosoftProjectStd2002.msi

Error Details

Item	Details
Test Name	Restart Manager Files In Use Dialog / Error Number 1
Description	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista.
Status	Ran
Details	Checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.
Help	TestHelpTopics\Restart Manager Files In Use Dialog.htm


Restart Manager Files In Use Dialog
Checks for Presence of MsiRMFilesInUse Dialog

Description

The Restart Manager Files In Use Dialog test checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.

Error Message

If this test detects that the MsiRMFilesInUse dialog is present in the package, the following message is displayed:

The 'Restart Manager Files In Use Dialog' test ran, but did not detect any problems as the 'MsiRMFilesInUse' dialog is present in the MSI.

Resolution: Automatic

PackageExpert automatically resolves this error by adding the MsiRMFilesInUse dialog to the package. In the output, the following message is listed:

The 'MsiRMFilesInUse' dialog was added to the MSI.

Figure 14-10: Message Node Detailed Information

Publishing Test Results to the Application Catalog

Every time you test a package in PackageExpert, the results are stored in a local .xml file in the same location as the .msi file.

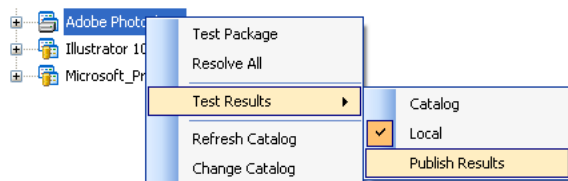
Name	Size	Type
program files		File Folder
Illustrator 10.msi	1,058 KB	Windows Installer P...
Illustrator 10.xml	26 KB	%XML Document

When testing an Application Catalog package, you can choose to publish the package's test results in the local .xml file to the Application Catalog.



Task: *To publish test results to an Application Catalog:*

1. In the Application Catalog package tree, select a package.
2. Right-click, point to **Test Results**, and click **Publish Results** on the context menu. You are prompted to confirm your selection.



3. Click **OK**. The local test results for that package are saved to the Application Catalog.

Publishing Test Results When Prompted

If you have tested an Application Catalog package, and you have not yet published its test results, when you disconnect from that Application Catalog or exit PackageExpert, the **Unpublished Packages** dialog box opens, prompting you to publish the test results:

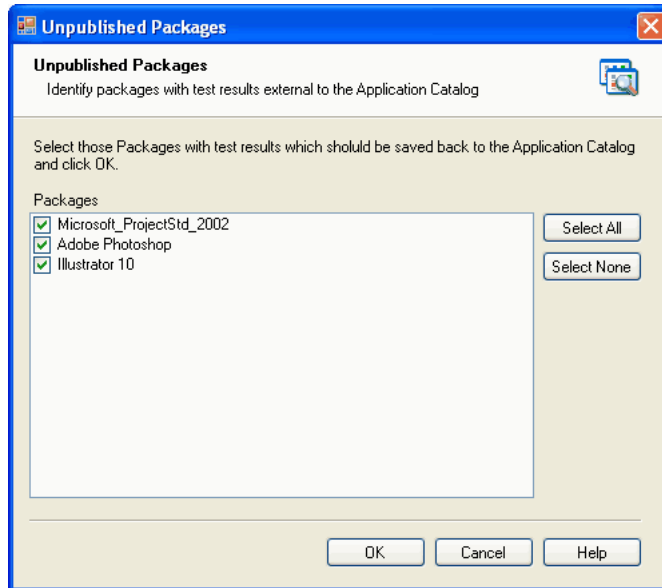


Figure 14-11: Prompt to Publish Unpublished Test Results

You are also prompted to publish test results if you disconnect from an Application Catalog and reconnect to it during the same PackageExpert session.



You can choose to publish the test results for all packages or only those packages that you select.

Toggling Between Local and Published Test Results

When viewing test results for Application Catalog packages, you can toggle between the locally saved test results (**Local**) and the results that have been published to the Application Catalog (**Catalog**).

The package icon used in the package tree varies depending upon which test results are being displayed:

Table 14-3 • Package Icons Indicating Test Result Mode

Icon	Mode	Results Being Displayed
	Local	Locally saved test results are being displayed.
	Catalog	Test results that were published to the Application Catalog are being displayed.

Chapter 14: Identifying and Resolving Package Errors Using PackageExpert

Testing Packages and Viewing Results

If PackageExpert is currently displaying the **Catalog** test results, but the **Catalog** results are out of sync with the results in the **Local** .xml file, an exclamation mark appears on the package icon:

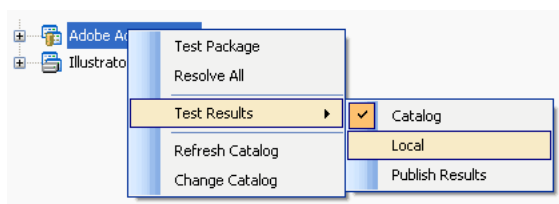


To toggle the PackageExpert test result display between Local and Catalog, perform the following steps.



Task: *To toggle the display of test results between Local and Catalog:*

1. In the Application Catalog package tree, select a package.
2. Right-click, point to **Test Results**, and click **Local** or **Catalog** on the context menu.



The selected test results are displayed and the package icon changes to indicate the mode you chose.

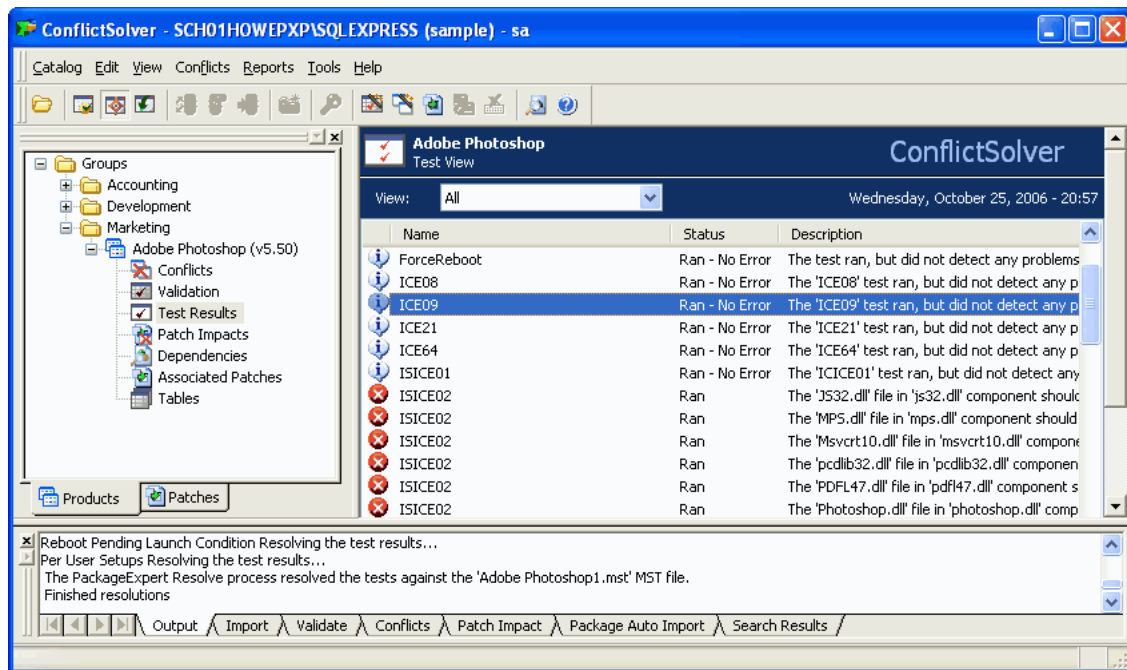
Viewing Test Results in ConflictSolver

You can test packages from the ConflictSolver interface, and view a summary of the results in the **Test Results** view. The **Test Results** view also displays a package's test results when a package is tested from the PackageExpert interface and the test results are published to the Application Catalog.



Note • By default, all functionality for interacting with the Application Catalog, including conflict detection and resolution, is available in Application Manager. Unless the **Turn off display of conflict detection and resolution options in Application Manager** option on the **General** tab of the **Options** dialog box is selected, you can also test packages and view test results using the Application Manager interface. See [Application Manager and ConflictSolver](#) for more information.

You open the **Test Results** view by selecting the **Test Result** node in the ConflictSolver package tree.



When the **Test Results** node is selected, any results that were previously published to the Application Catalog regarding this package are displayed.

This list of messages can be filtered by **All**, **Ran - No Error**, **Ran**, or **Resolved** by making a selection from the **View** list.

Deleting Test Results

You can delete a package's local test results and, for packages in the Application Catalog, the test results published to the Application Catalog.

Deleting a Package's Local Test Results

Every time you test a package in PackageExpert, the results are stored in a local **.xml** file in the same location as the **.msi** file. If you want to delete these test results, delete the **.xml** file.

Name	Size	Type
program files		File Folder
Illustrator 10.msi	1,058 KB	Windows Installer P...
Illustrator 10.xml	26 KB	XML Document

Deleting a Package's Published Test Results

You can delete PackageExpert test results that have been published to the Application Catalog using the Application Manager interface.

To delete a package's published test results, perform the following steps:



Task: *To delete a package's published test results:*

1. Open Application Manager.
2. Select a package in the package tree, point to **Delete** and click **Persisted Test Information** from the context menu. The PackageExpert test results for that package are deleted from the Application Catalog.

Customizing Test Results

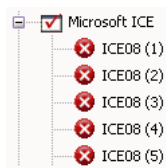
You are able to customize PackageExpert test results to accommodate unique or special circumstances at your organization by changing the severity of messages and by suppressing the display of a specific instance of an error.

- [Changing the Severity of a Specific Error](#)
- [Suppressing the Display of Specific Errors](#)

Changing the Severity of a Specific Error

Using the **Test Result Filter Editor**, you can change the severity of a specific instance of a message from its original severity to **Error**, **Warning**, or **Information**.

You are changing the severity of a *specific instance* of a message generated by a test, not the severity of all messages generated by that test. For example, suppose the ICE08 test produced five error messages:



You consider messages 1 through 4 to be valid errors. However, suppose that message 5 concerns a particular component of the package that no one at your organization is likely to use, and therefore you do not want this particular error message to have a severity of **Error**. You can change the severity to **Warning** to document the decreased importance of this particular message at your organization.

Error Details

Item	Details
Test Name	ICE08 / Error Number 5
Description	The 'Global_RegFiles_GifRepair' was provided with a new ComponentId.
Status	Resolved
Details	Validates that the Component table contains no duplicate Component IDs (GUIDs).
Help	TestHelpTopics\Microsoft ICE_ICE08.htm

Figure 14-12: Error Details of Error Number 5 of the ISICE08 Test

Information about changing the severity of a message is presented in the following topics:

- [Changing the Severity of a Message](#)
- [Editing a Result Filter](#)
- [Deleting a Result Filter](#)

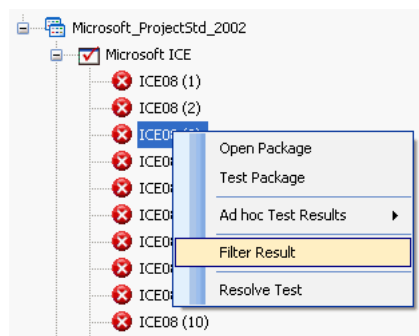
Changing the Severity of a Message

To change the severity of a message, perform the following steps:



Task: *To change the severity of a message:*

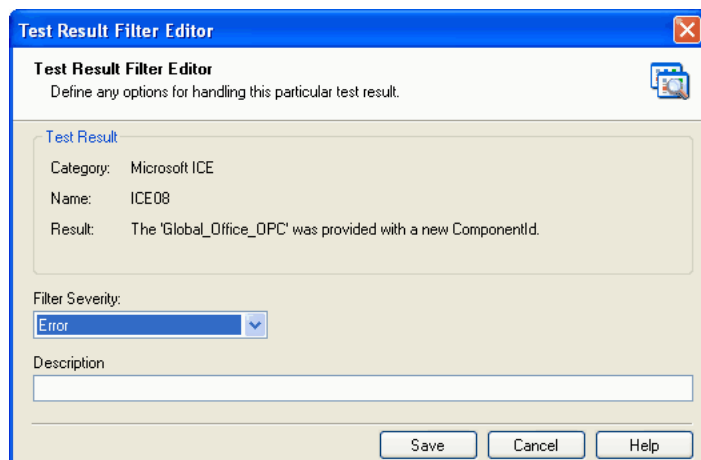
1. On the **Test & Resolve** view, select the specific message that you want to modify and select **Filter Result** from the context menu.



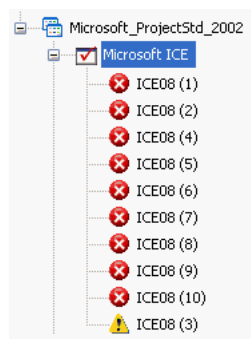
The **Test Result Filter Editor** dialog box opens:

Chapter 14: Identifying and Resolving Package Errors Using PackageExpert

Testing Packages and Viewing Results



2. From the **Filter Severity** list, select the severity that you want to change this message to: **Error**, **Warning**, or **Information**.
3. In the **Description** box, enter an explanation for why you changed the severity level of this message.
4. Click **Save**. This error now displays the icon of its new severity level:



If the same problem is encountered in another package or in a subsequent test run of this package, the severity level that you defined will continue to be used.

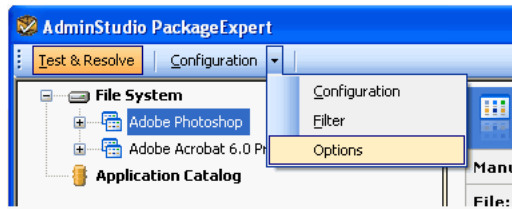
Editing a Result Filter

To edit an existing filter, perform the following steps.

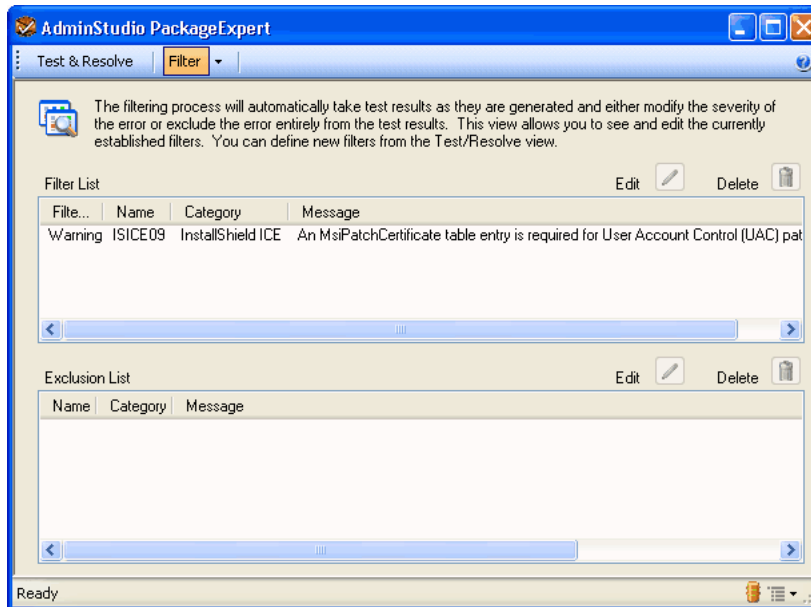


Task: *To edit an existing filter:*

1. In the PackageExpert toolbar, click the down arrow next to the second menu item and select **Filter**.



The **Filter** view opens, listing all of the defined severity filters in the **Filter List**:



2. Select the filter that you want to edit, and click **Edit**. The **Test Result Filter Editor** dialog box opens.
3. Edit the **Filter Severity** and/or **Description** fields and click **Update** to save your edits.

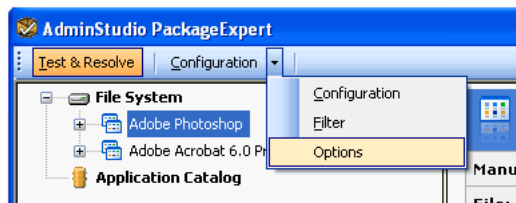
Deleting a Result Filter

To delete a filter, so that the a message will either have its default severity level or will return to the list after being excluded, perform the following steps.

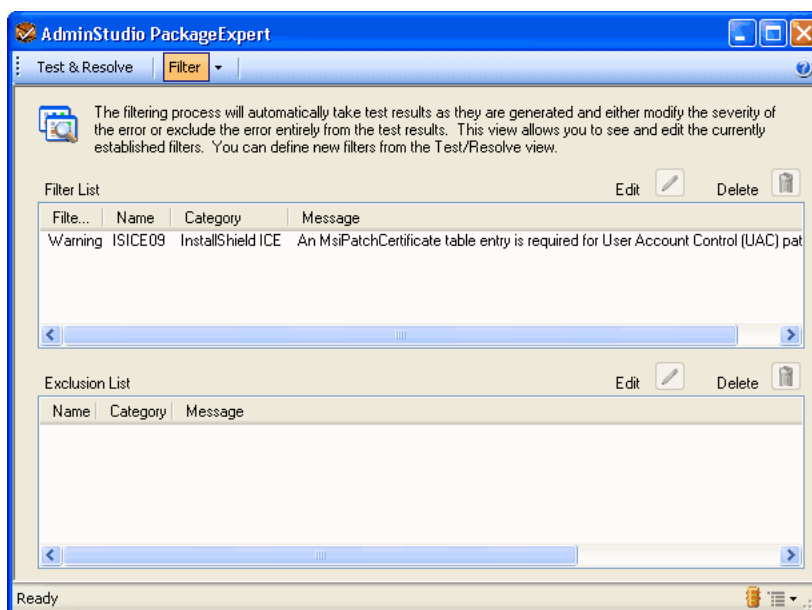


Task: *To delete a result filter and reset to original severity:*

1. In the PackageExpert toolbar, click the down arrow next to the second menu item and select **Filter**.



The **Filter** view opens, listing all of the defined severity filters in the **Filter List**:

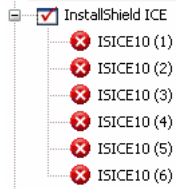


2. Select the filter that you want to delete, and click **Delete**. You are prompted to confirm the deletion.
3. Click **OK**.

Suppressing the Display of Specific Errors

Using the **Test Result Filter Editor**, you can choose to exclude the display of a *specific instance* of a message.

You are suppressing a *specific instance* of a message generated by a test, not suppressing all messages generated by that test. For example, suppose the ISICE10 test produced six error messages:



You consider messages 1, 2, 4, 5, and 6 to be valid errors. However, suppose that message 3 concerns a particular component of the package (ISSelfRegisterCosting in the example below) that is no longer used at your organization, and therefore you want to suppress the display of this particular error message.

Error Details

Item	Details
Test Name	ISICE10 / Error Number 3
Description	The custom action 'ISSelfRegisterCosting' is not documented in table 'ISCustomActionReference'.
Status	Ran
Details	Verifies that each custom action in your installation is documented by validating that each entry in the CustomAction table has a corresponding ISCustomActionReference table entry.
Help	TestHelpTopics\InstallShield ICE_ISICE10.htm

Figure 14-13: Error Details of Error Number 3 of the ISICE10 Test

Information about suppressing the display of an error message is presented in the following topics:

- [Suppressing the Display of a Message](#)
- [Editing a Result Filter](#)
- [Deleting a Result Filter](#)

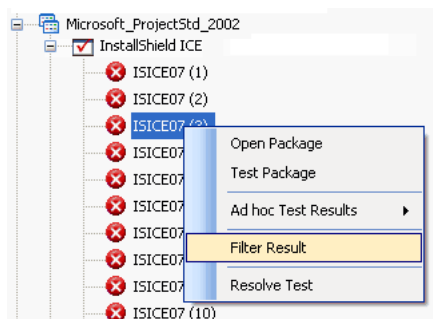
Suppressing the Display of a Message

To suppress the display of a message, perform the following steps:

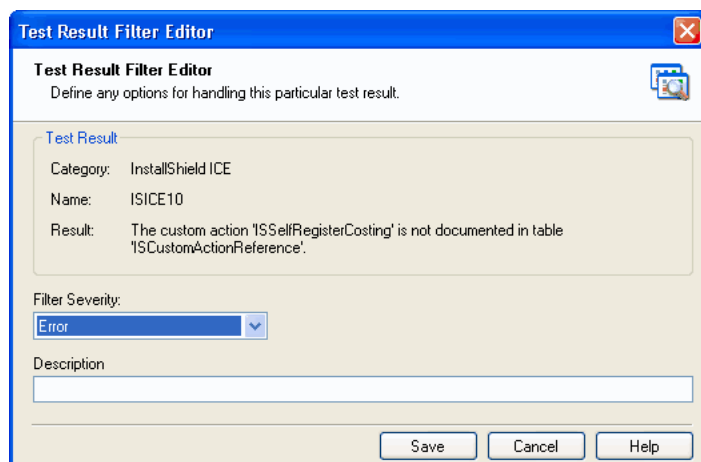


Task: *To suppress the display of a message:*

1. On the **Test & Resolve** view, select the specific message that you want to suppress and select **Filter Result** from the context menu.



The **Test Result Filter Editor** dialog box opens:



2. From the **Filter Severity** list, select **Exclude**.
3. In the **Description** box, enter an explanation for why you chose to exclude this message.
4. Click **Save**. The message is no longer listed in the package tree

If the same problem is encountered in another package or in a subsequent test run of this package, the error message will not be displayed.

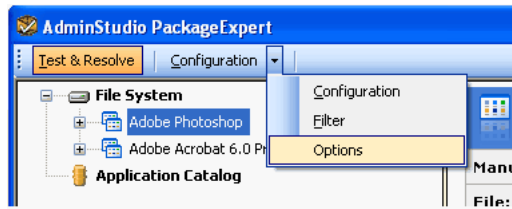
Editing a Result Filter

To edit an existing filter, perform the following steps.

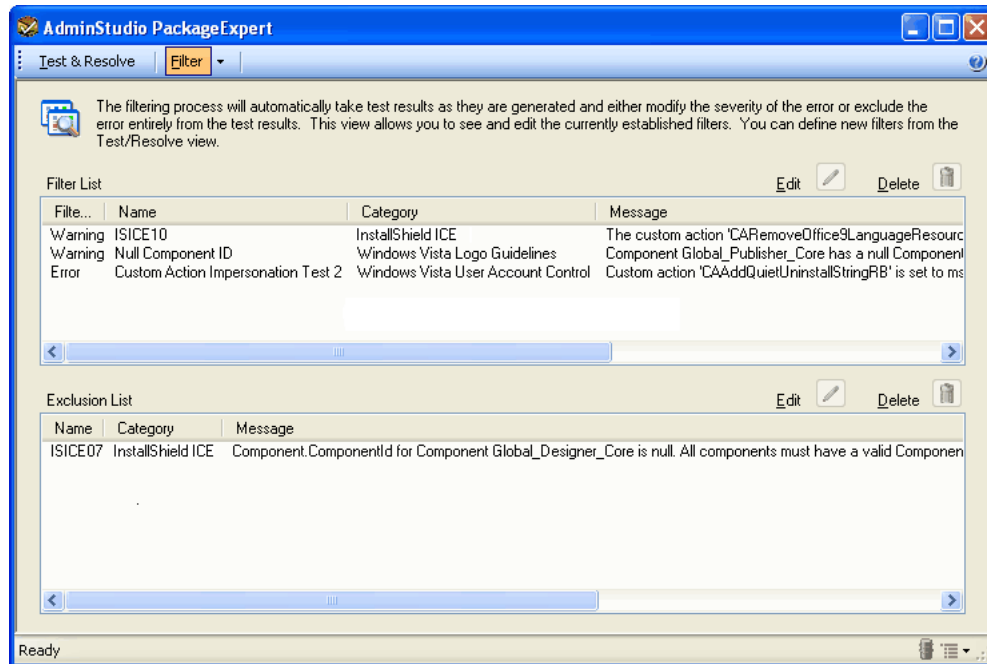


Task: *To edit an existing filter:*

1. In the PackageExpert toolbar, click the down arrow next to the second menu item and select **Filter**.



The **Filter** view opens, listing all of the defined Exclude filters in the **Exclusion List**:



2. Select the filter that you want to edit, and click **Edit**. The **Test Result Filter Editor** dialog box opens.
3. Edit the **Filter Severity** and/or **Description** fields and click **Update** to save your edits.

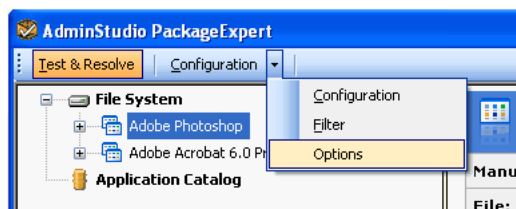
Deleting a Result Filter

To delete a filter, so that this message will no longer be excluded, perform the following steps.

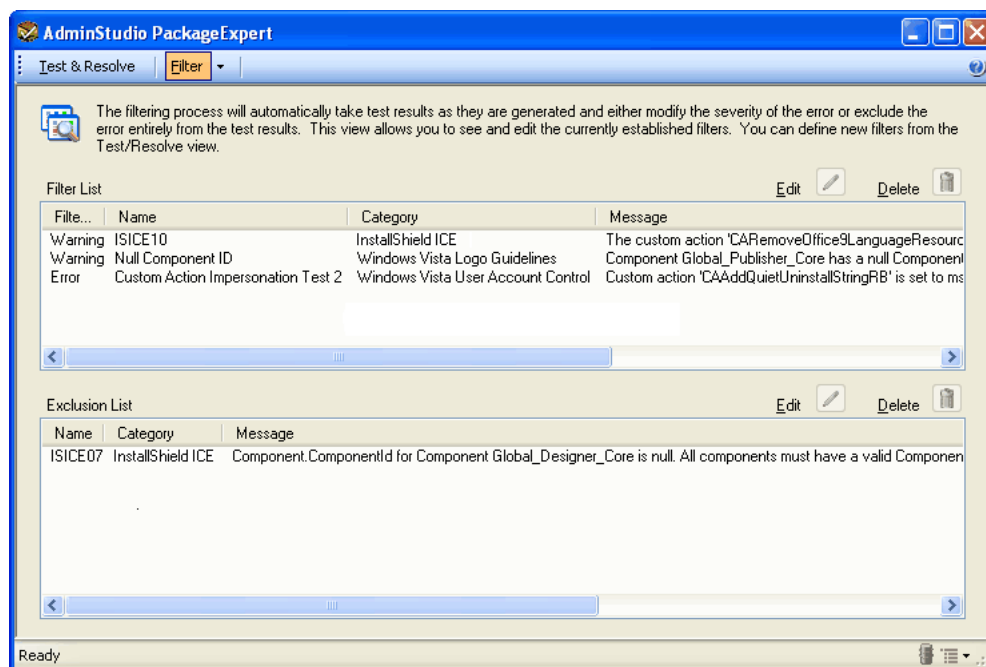


Task: *To delete a result filter and reset to original severity:*

1. In the PackageExpert toolbar, click the down arrow next to the second menu item and select **Filter**.



The **Filter** view opens, listing all of the defined exclusion filters in the **Exclusion List**:



2. Select the filter that you want to delete, and click **Delete**. You are prompted to confirm the deletion.
3. Click **OK**.

Adding Ad-Hoc Test Results

You can manually add a test result message to a package. You might want to do this if you tested the package using QualityMonitor, InstallShield, or another tool, and you noticed something which is unexpected and which you want to document. There might be other scenarios such as an installation failure when you might want to document an error.

The following example documents an error was encountered during installation, which caused installation failure.

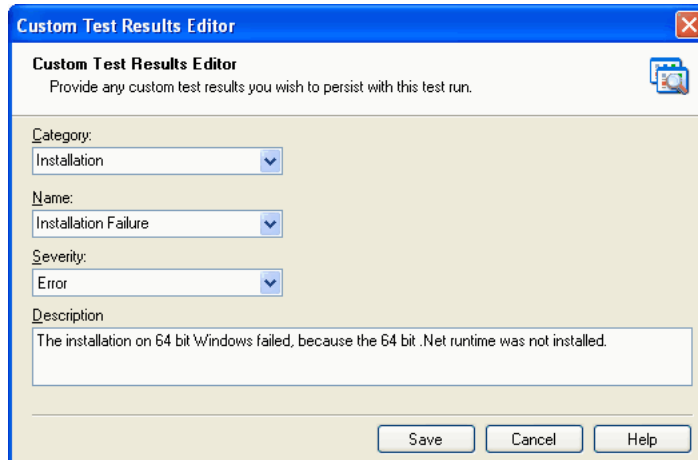


Figure 14-14: Example of a Message Added to Document Ad-Hoc Testing

When you click **Save** on the **Custom Test Results Editor** dialog box, the new message will be displayed in the package tree, and will be documented in the **Error Details** area in the right pane, as shown in the following Figure:

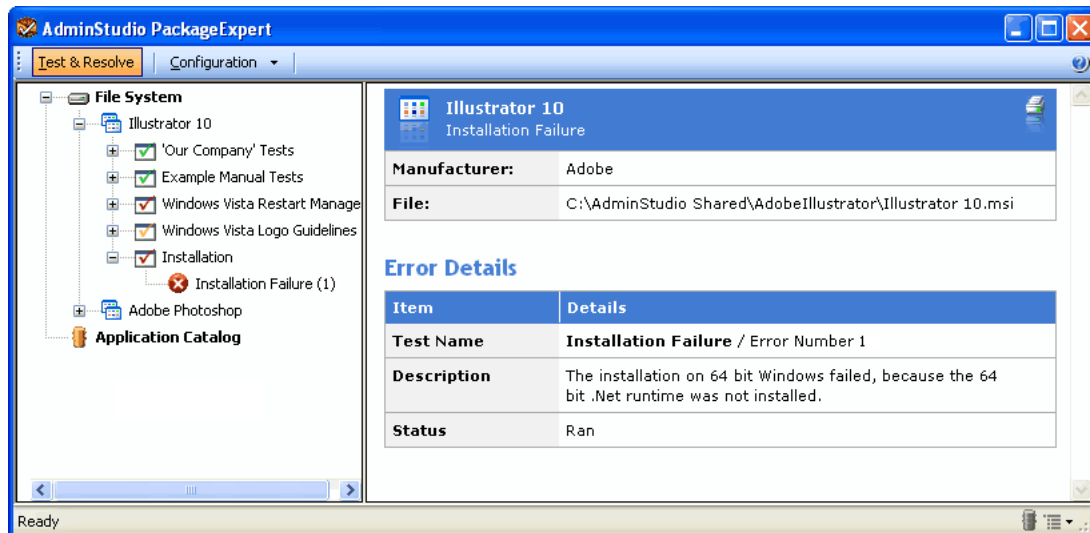


Figure 14-15: Display of a Message Added to Document Ad-Hoc Testing

When this problem is corrected, you could then run the installation on 64 bit Windows, confirm the problem is corrected, and delete or demote the severity of the test result.

The following procedures are provided:

- [Adding a Test Result Message](#)
- [Editing a Test Result Message](#)
- [Deleting a Test Result Message](#)

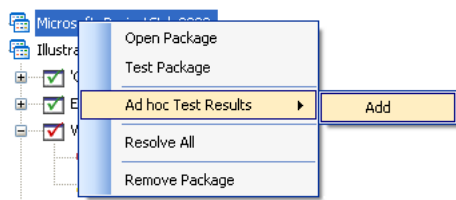
Adding a Test Result Message

To add a test result message, perform the following steps.



Task: *To add a test result message to a package:*

1. Select the package in the package tree that you want to add a test result message to.
2. Right-click on the package, point to **Ad hoc Test Results** and click **Add** on the context menu. The **Custom Test Results Editor** dialog box opens.



3. In the **Category** list, either enter a new name for a test category where you want to display this message, or select an existing category from the list.



Note • Custom test results only reside in custom test result categories (test categories that you created). If a category has already been added, then it will be listed in the **Category** list. You can create a new category by entering a category name in the **Category** box.

4. In the **Name** list, either enter a new name for the test, or—if the same test has generated more than one error—select an existing name from the list.
5. From the **Severity** list, select **Error**, **Warning**, or **Information**.
6. In the **Description** box, document the error that you discovered during ad-hoc testing. This information will be displayed under **Error Details** in the right pane when this test result node is selected.

Editing a Test Result Message

To edit the **Severity** or **Description** of an existing test result message, perform the following steps.



Task: *To edit a test result message:*

1. Select the message node in the package tree that you want to edit.
2. Right-click on the message node, point to **Ad hoc Test Results** and click **Edit** on the context menu. The **Custom Test Results Editor** dialog box opens, with the **Severity** and **Description** fields enabled.
3. Make edits to the **Severity** and **Description** fields and click **Update**.

Deleting a Test Result Message

To delete an existing test result message, perform the following steps.



Task: To delete a test result message:

1. Select the message node in the package tree that you want to delete.
2. Right-click on the message node, point to **Ad hoc Test Results** and click **Delete** on the context menu. You are prompted to confirm the deletion.
3. Click **OK**.

Resolving Errors

For some tests, PackageExpert can automatically resolve any errors that are found. PackageExpert will either modify the .msi package or create a transform (.mst) file to apply the changes, depending upon what you have specified on the **Options** view, as described in [Specifying the Resolution Method](#).

You can choose to resolve all errors for a package, all errors in a Test Category, or just one specific error.



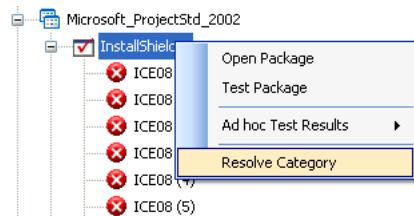
Task: To automatically resolve errors:

1. In the **Test & Resolve** view, review the messages generated during testing for a package.



Note • To determine whether a test is automatically resolvable, select the message node, and check to see whether **Resolution: Automatic** or **Resolution: Manual** is displayed in the help topic in the right pane.

2. Select a package, a test category, or a message node, and select **Resolve All**, **Resolve Category**, or **Resolve Test** from the context menu



Messages appear in the Output window informing you of each resolution. When the resolutions are complete, the name of the file that was edited is listed: either a transform file or the .msi package, as specified on the **Configuration** tab. For example:

```
Resolution started...
ICE08 Resolving the test results...
ICE08 Resolving the test results...
ICE08 Resolving the test results...
```

ICE08 Resolving the test results...

ICE09 Resolving the test results...

ICE09 Resolving the test results...

ICE09 Resolving the test results...

ICE21 Resolving the test results...

The PackageExpert Resolve process resolved the tests against the 'MicrosoftProjectStd2002.mst' MST file.

Finished

When a Warning or Error message is resolved, a green check mark is listed next to that message in the Test Results list:

Test Results for Category: Microsoft ICE

	Name	Status	Description	Help
✓	ICE08	Resolved	The 'Global_Authoring_OPCLRemoveMax' was provided with a new ComponentId.	Info
✓	ICE08	Resolved	The 'Global_Authoring_OPCLRemoveMin' was provided with a new ComponentId.	Info
✓	ICE08	Resolved	The 'Global_Fonts_Tahoma' was provided with a new ComponentId.	Info
✓	ICE08	Resolved	The 'Global_Office_OPC' was provided with a new ComponentId.	Info
✓	ICE08	Resolved	The 'Global_Office_OPCLData' was provided with a new ComponentId.	Info

Adding Manual Tests

In PackageExpert, you can add a Manual Test to the package tree by creating an .xml file and placing it in the PackageExpert installation directory. By adding Manual Tests to Package Expert, you can document important tests at your organization and monitor whether these tests are performed.

- [About Manual Tests](#)
- [Creating a Manual Test](#)
- [Running and Resolving a Manual Test](#)
- [Sample Manual Test](#)

About Manual Tests

A Manual Test is an .xml file that contains both instructions (or links to those instructions) on exactly how to perform a Manual Test to evaluate the quality of a Windows Installer package, and instructions on how to resolve any errors that are found.

PackageExpert provides two sample Manual Tests, **Desktop Icons** and **Example**, which are displayed in the package tree:

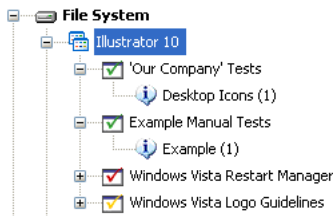


Figure 14-16: Manual Tests Listed in the Package Tree

You perform or resolve a Manual Test by selecting the test and making a selection from the context menu: **Perform Manual Test** or **Resolve Manual Test**.

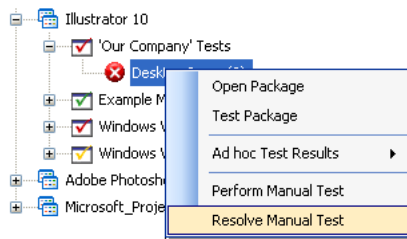


Figure 14-17: Manual Test Selections on the Context Menu

When you select **Perform Manual Test**, the instructions defined in the .xml file are listed on the **Manual Test Perform** dialog box. As you perform each step, you mark each one **Pass** or **Fail**.

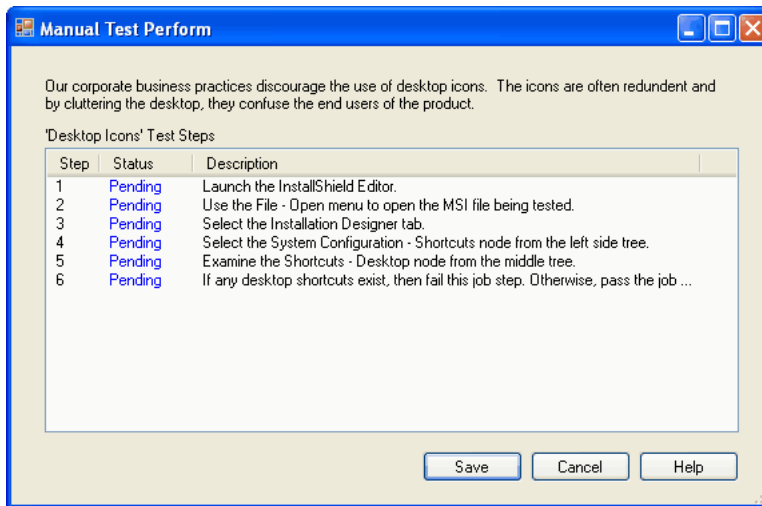


Figure 14-18: Steps to Perform a Manual Test

After you complete all of the steps on the **Manual Test Perform** dialog box, the **Resolve Manual Test** selection is enabled, and the instructions on how to resolve errors discovered by this test are listed on the **Perform Manual Test** dialog box:

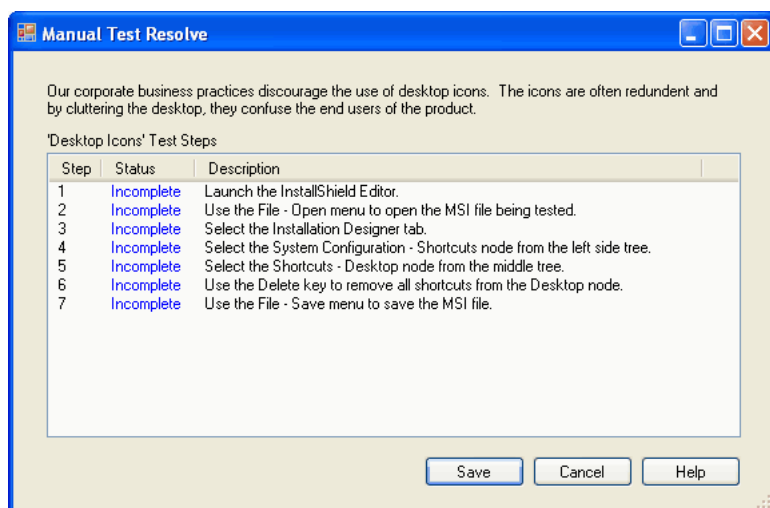


Figure 14-19: Steps to Resolve a Manual Test

To help you learn how to construct your own Manual Test, examine the sample .xml file provided by PackageExpert. This file, ManualTest.xml, can be found in the PackageExpert directory inside the AdminStudio installation directory.

For information on the structure of a Manual Test in an .xml file, see [Sample Manual Test](#).

Creating a Manual Test

By adding Manual Tests to Package Expert, you can document important tests at your organization and monitor whether these tests are performed.

A Manual Test is an .xml file that contains both instructions (or links to those instructions) on exactly how to perform a Manual Test to evaluate the quality of a Windows Installer package, and instructions on how to resolve any errors that are found.

To write your own Manual Test, perform the following steps:



Task: *To write a Manual Test:*

1. Locate and open the following file in a code editor:

C:\Program Files\AdminStudio\9.5\PackageExpert\manualtest.xml

2. Locate and make a copy of a <TestCase> element in that XML file:

```
<TestCase name="Example" category="Example Manual Tests" class=""
testtype="Manual" selected="false">
  <TestResults>
    <TestResult state="Not run" severity="Informational" type="Manual" ordinal="1">
      The 'Example' test has not yet been run.
    <TestData></TestData></TestResult>
  </TestResults>
```

```
<TestData>
  <Details>The 'Example' test is run to detect the state of something and respond
  accordingly. You should describe why it is important.</Details>
  <KBArticle>TestHelpTopics\Example.htm</KBArticle>
  <Perform>
    <Incomplete state="Ran - Incomplete" severity="Informational">The 'Example' test has
    not been completely run.</Incomplete>
    <Fail state="Ran" severity="Informational">The 'Example' test ran and failed.</Fail>
    <Pass state="Ran" severity="Error">The 'Example' test ran successfully.</Pass>
    <Steps>
      <Step step="1" state="Pending" type="text">Step 1 - Describe the process with
      inline text.</Step>
      <Step step="2" state="Pending" type="link">Step 2 - Instructions on a web page.
      <Direct>http://www.cnn.com</Direct></Step>
      <Step step="3" state="Pending" type="file">Step 3 - Instructions referencing a
      local file.<Direct>images\info.gif</Direct></Step>
    </Steps>
  </Perform>
  <Resolve>
    <Incomplete state="Ran" severity="Warning">The 'Example' test resolution process is
    incomplete.</Incomplete>
    <Complete state="Resolved" severity="Informational">The 'Example' test resolution
    process completed successfully.</Complete>
    <Steps>
      <Step step="1" state="Incomplete" type="text">Step 1 - Describe the resolution
      process with inline text.</Step>
      <Step step="2" state="Incomplete" type="link">Step 2 - Resolution instructions on a
      web page.<Direct>http://www.msnbc.com</Direct></Step>
    </Steps>
  </Resolve>
</TestData>
</TestCase>
```

3. Customize the copy of the <TestCase> element to define your Manual Test, as described in [Sample Manual Test](#).
4. Open PackageExpert. If, in the <TestCase> element you set selected="true", a new node will be listed in the Test Category you entered.

Running and Resolving a Manual Test

PackageExpert provides a sample Manual Test named **Desktop Icons** that you can use to learn how to run and resolve a Manual Test.

Running a Manual Test

To run the **Desktop Icons** Manual Test, perform the following steps:

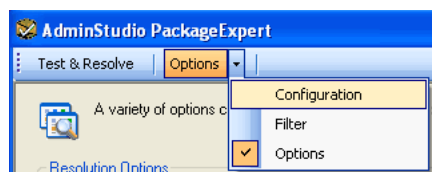


Task: **To run this test:**

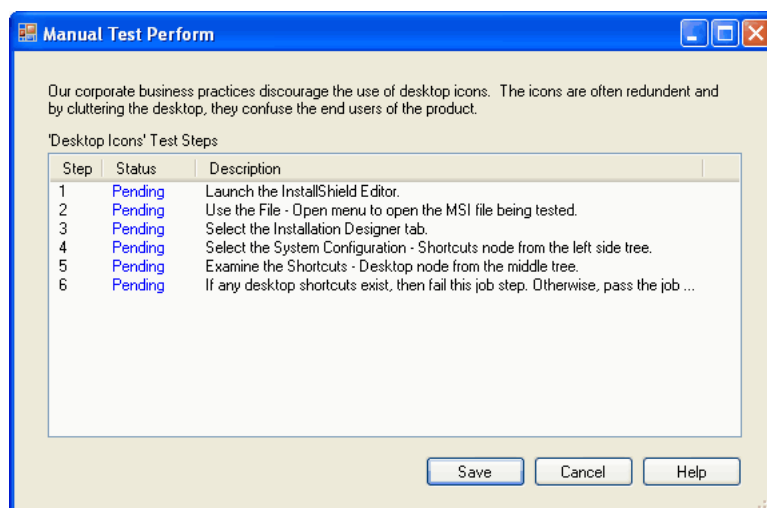
1. Click **Configuration** in the PackageExpert toolbar. The **Configuration** view opens.



Note • The second menu item in the PackageExpert toolbar toggles between **Configuration**, **Options**, and **Filter**. If **Configuration** is not displayed, click the arrow next to the menu item and select **Configuration** from the list.



2. Select the manual test that you want to run and resolve.
3. Return to the **Test & Resolve** view and open a package in PackageExpert.
4. Select the **Desktop Icons** node and select **Perform Manual Test** from the context menu. The **Manual Test Perform** dialog box opens, listing the manual steps you need to perform to run this manual tests.



5. Perform each of the steps, and mark each as **Pass** or **Fail** after you have performed it by clicking on the step and selecting **Pass** or **Fail** from the menu.
6. Click **Save**.

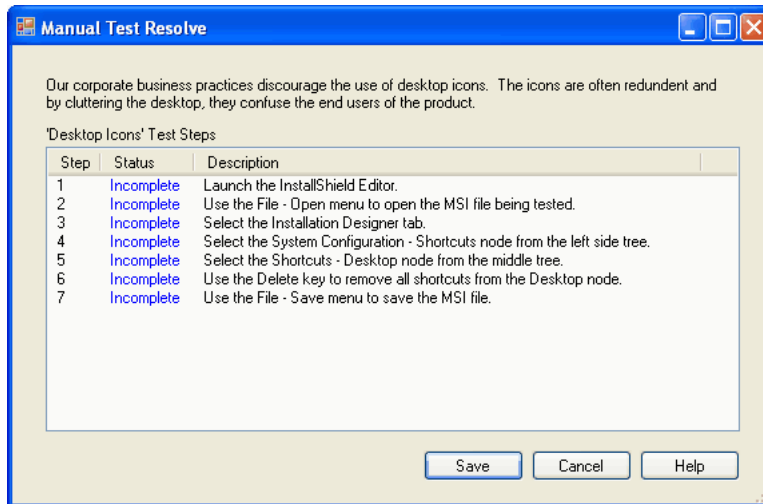
Resolving a Manual Test

When performing a manual test, you mark the **Status** of each step as either **Pass**, **Fail**, or **Pending**. If you have set the status of any of the steps to **Fail**, and none of the steps are still in **Pending** status, the **Resolve Manual Test** selection on the context menu is enabled, and you can proceed with resolution.



Task: *To resolve this test:*

1. With the package open in PackageExpert, select the **Desktop Icons** node and select **Resolve Manual Test** from the context menu. The **Manual Test Resolve** dialog box opens, listing the manual steps you need to perform to resolve this error.



2. Perform each of the steps, and mark each as **Complete** after you have performed it by clicking on the step and selecting **Complete** from the menu.
3. Click **Save**.
4. Select the **Desktop Icons** node. In the **Error Details** that are displayed, this test's **Status** is now **Resolved**, and the **Description** is *The test resolution process is complete and the desktop icons removed.*

Sample Manual Test

PackageExpert includes two sample Manual Tests.

- **Example of a Manual Test**—The **Desktop Icons** Manual Test is a functional example of a Manual Test. It examines a Windows Installer package for shortcuts destined for the Desktop and removes those shortcuts.
- **Template for a Manual Test**—The **Example** manual test is a template that you can use to create your own manual test. It is intended to illustrate the possible functionality and authoring requirements for end-user Manual Tests. Its structure is described below.

The **Example** Manual Test has the following code:

```
<TestCase name="Example" category="Example Manual Tests" class=""
testtype="Manual" selected="false">
  <TestResults>
    <TestResult state="Not run" severity="Informational" type="Manual" ordinal="1">
      The 'Example' test has not yet been run.
    <TestData></TestData></TestResult>
```

```

</TestResults>
<TestData>
  <Details>The 'Example' test is run to detect the state of something and respond
  accordingly. You should describe why it is important.</Details>
  <KBArticle>TestHelpTopics\Example.htm</KBArticle>
  <Perform>
    <Incomplete state="Ran - Incomplete" severity="Informational">The 'Example' test has
    not been completely run.</Incomplete>
    <Fail state="Ran" severity="Informational">The 'Example' test ran and failed.</Fail>
    <Pass state="Ran" severity="Error">The 'Example' test ran successfully.</Pass>
    <Steps>
      <Step step="1" state="Pending" type="text">Step 1 - Describe the process with
      inline text.</Step>
      <Step step="2" state="Pending" type="link">Step 2 - Instructions on a web page.
      <Direct>http://www.cnn.com</Direct></Step>
      <Step step="3" state="Pending" type="file">Step 3 - Instructions referencing a
      local file.<Direct>images\info.gif</Direct></Step>
    </Steps>
  </Perform>
  <Resolve>
    <Incomplete state="Ran" severity="Warning">The 'Example' test resolution process is
    incomplete.</Incomplete>
    <Complete state="Resolved" severity="Informational">The 'Example' test resolution
    process completed successfully.</Complete>
    <Steps>
      <Step step="1" state="Incomplete" type="text">Step 1 - Describe the resolution
      process with inline text.</Step>
      <Step step="2" state="Incomplete" type="link">Step 2 - Resolution instructions on a
      web page.<Direct>http://www.msnbc.com</Direct></Step>
    </Steps>
  </Resolve>
</TestData>
</TestCase>

```

Each element of this code defines a part of the Manual Test:

Table 14-4 • XML Elements of a Manual Test

Element	Purpose
<TestCase>	Begins the Manual Test code and identifies the test name and category. Also, if selected="true" in this element, this test will be listed in the package tree on the Test & Resolve view.
<TestResults>	Lists the information that will be displayed in the Description field of the Error Details table if this test has not yet been run.
<TestData>	Marks the beginning and the end of the sections containing the instructions for performing and resolving the test.
<Details>	Lists the information that will be displayed in the Details field of the Error Details table.

Table 14-4 • XML Elements of a Manual Test

Element	Purpose
<KBArticle>	Contains a link to a help topic that provides more information on this test. This list appears in the Help field of the Error Details table.
<Perform>	Marks the beginning and the end of the instructions to perform this test.
<Incomplete state> <Pass state> <Fail state>	Identifies the states that will be displayed when you click on a step on the Perform Manual Test dialog box, the severity of each state, and the text that will be displayed when this state is selected.
<Steps>	Marks the beginning and the end of the steps that have to be completed to perform this test.
<Step>	Marks the beginning and the end of each step in this test.
<Resolve>	Marks the beginning and the end of the instructions to resolve this test.
<Incomplete state> <Complete state>	Identifies the states that will be displayed when you click on a step on the Resolve Manual Test dialog box, the severity of each state, and the text that will be displayed when this state is selected.
<Steps>	Marks the beginning and the end of the steps that have to be completed to resolve this test.
<Step>	Marks the beginning and the end of each step in this resolution.

PackageExpert Reference

PackageExpert Reference includes the same topics displayed when you click a help button from a view or dialog box in the PackageExpert interface. Reference information for PackageExpert is organized into the following area.

Table 14-5 • PackageExpert Reference

Section	Description
PackageExpert Interface	This section contains information about the various views of the PackageExpert interface and how they work together.
PackageExpert Views	Each view available in PackageExpert is covered in this section.
PackageExpert Dialog Boxes	Specific help for each dialog box in PackageExpert is covered in this section.

PackageExpert Interface

This section describes the overall PackageExpert user interface and includes the following topics:

- [PackageExpert Home Page](#)
- [PackageExpert Package Tree](#)
- [PackageExpert Toolbar Menus](#)
- [PackageExpert Icons](#)
- [Status Bar](#)

PackageExpert Home Page

If no packages are open, the initial view of the PackageExpert interface is the PackageExpert Home page. On the Home Page, the following quick links are provided:

- **Open a Package**—Open a Windows Installer package to test. See [Testing External Packages](#).
- **Open a Catalog**—Open an Application Catalog containing packages to test. See [Testing Packages in an Application Catalog](#).
- **Set Defaults**—Select the default tests to run each time a package is tested, and specify the resolution method you want to use. See [Setting Default Test Configuration](#) and [Specifying the Resolution Method](#).

PackageExpert Package Tree

The PackageExpert package tree includes the following nodes:

Table 14-6 • PackageExpert Package Tree

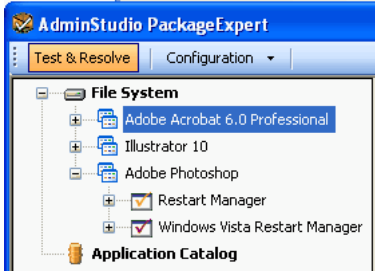
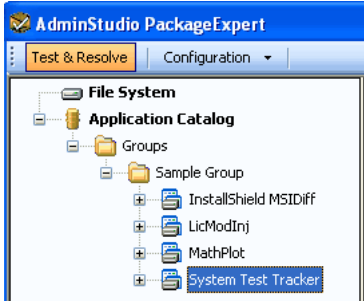
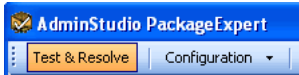
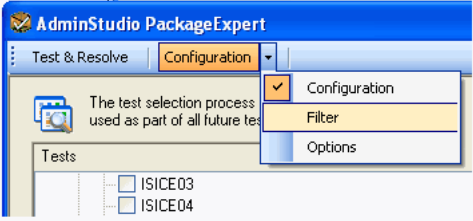
Item	Description
File System Node	<p>Select this node and select Open Package from the context menu to open a standalone Windows Installer package to test. All of the packages that are currently open as listed under the File System node.</p>  <p>Note • When you open a package under the File System node, it remains open through multiple PackageExpert sessions, until you explicitly remove it. To remove a package from the list, select it and select Remove Package from the context menu.</p>
Application Catalog Node	<p>When you select this node, you are prompted to connect to an Application Catalog. After you have connected to an Application Catalog, the groups and packages it contains are listed in a tree format under the Application Catalog node:</p> 
Test & Resolve Menu	<p>Click this menu to return to the Test & Resolve page from a configuration page.</p> 

Table 14-6 • PackageExpert Package Tree

Item	Description
Configuration / Filter / Options Menu	<p>Click the arrow in the link to toggle between the following views:</p>  <ul style="list-style-type: none">• Configuration—Select to open the Configuration View, where you specify which tests you want to run for all subsequent test runs.• Filter—Select to open the Filter View, where you can edit or delete existing filters.• Options—Select to open the Options View where you can specify whether you want PackageExpert to resolve errors by creating a transform (.mst) file or by directly editing the .msi package.

PackageExpert Toolbar Menus

The PackageExpert toolbar includes the following menu selections:

Table 14-7 • PackageExpert Package Tree


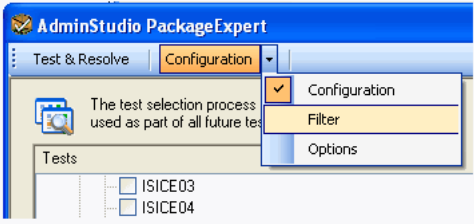
Item	Description
Test & Resolve Menu	<p>Click this menu to return to the Test & Resolve view from another view.</p> 

Table 14-7 • PackageExpert Package Tree

Item	Description
Configuration / Filter / Options Menu	<p>Click the arrow in the link to toggle between the following views:</p>  <ul style="list-style-type: none"> • Configuration—Select to open the Configuration View, where you specify which tests you want to run for all subsequent test runs. • Filter—Select to open the Filter View, where you can edit or delete existing filters. • Options—Select to open the Options View where you can specify the resolution method.

PackageExpert Icons





There are two types of PackageExpert icons:

- [Message Severity Icons](#)
- [Package Tree Icons](#)

Message Severity Icons

When you select a Test Category node, all of the messages that were generated by tests in that category are listed. Each message includes an icon to indicate its severity. The severity icons indicate the following:

Table 14-8 • Severity Icons

Icon	Status	Description
	Error	Identifies a problem that would cause incorrect behavior when this package is installed or used.
	Warning	Identifies a problem that could cause incorrect behavior when this package is installed or used.
	Information	Indicates that either the test has not yet been run or the test ran and generated no errors.
	Resolved	Indicates that PackageExpert has repaired this error.

Package Tree Icons

The following icons are used in the package tree:

Table 14-9 • Package Tree Icons















Icon	Name	Description
	File System	Select this icon and select Open Package from the context menu to open and test a package that is stored on a local or network file system.
	Application Catalog	Select this icon to connect to an SQL Server Application Catalog database or to the AdminStudio Enterprise Server Application Catalog.
	Application Catalog Disabled	Appears in Status Bar when PackageExpert is not connected to an Application Catalog.
	Tests Category With Errors	Test results in this category include at least one message with a status of Error .
	Test Category With Warnings	Test results in this category includes at least one message with the status of Warning and no messages with the status of Error .
	Test Category With No Errors or Warnings	Either the tests in this category have not been run, or all messages in this category have a status of Information (there are no messages with the status of Error or Warning).
	Package	Represents a Windows Installer package.
	Package Displaying Catalog Test Results	<p>This package is in an Application Catalog and PackageExpert is currently displaying the PackageExpert test results that have been published to that Application Catalog.</p> <p>To view a package's published test results:</p>  <hr/> <p>To display a package's published test results:</p> <ol style="list-style-type: none"> 1. Select the package in the package tree. 2. Right-click, point to Test Results and click Catalog on the context menu.

Table 14-9 • Package Tree Icons

Icon	Name	Description
	Package Displaying Local Test Results	<p>This package is in an Application Catalog and PackageExpert is currently displaying the PackageExpert test results stored in this package's local .xml file.</p> <p></p> <hr/> <p>To display a package's local test results:</p> <ol style="list-style-type: none"> 1. Select the package in the package tree. 2. Right-click, point to Test Results and click Local on the context menu.
	Out of Sync Package	<p>This package is in an Application Catalog and PackageExpert is currently displaying the PackageExpert test results that have been published to that Application Catalog, but the Catalog results are out of sync with the results in the local .xml file</p> <p></p> <hr/> <p>To publish local test results to the Application Catalog:</p> <ol style="list-style-type: none"> 1. Select the package in the package tree. 2. Right-click, point to Test Results and click Publish Results on the context menu.
	Disabled Package	<p>For packages in an Application Catalog, this icon indicates either that the Windows Installer .msi file associated with this package has been moved or deleted from its original location, or that this package is managed by the Software Repository.</p>

Status Bar

A Status Bar appears at the bottom of all of the PackageExpert Views:

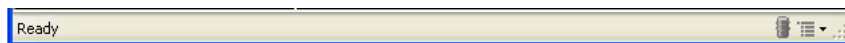


Figure 14-20: PackageExpert Status Bar

The Status Bar is comprised of the following items:

- PackageExpert Run Status
- Application Catalog Status
- Output Window Toggle

PackageExpert Run Status

PackageExpert has the following statuses:



Table 14-10 • PackageExpert Run Status

Status	Description
Ready	No tests or automated resolutions are currently being executed.
Executing automated tests on [PACKAGE_NAME]	PackageExpert is executing a test run.
Resolution started	PackageExpert is performing automated resolutions of the selected tests or test categories.

Application Catalog Status

An icon appears on the right side of the Status bar to indicate whether or not PackageExpert is currently connected to an Application Catalog:

Table 14-11 • Application Catalog Status Icons

Icon	Name	Description
	Connected to an Application Catalog	PackageExpert is connected to an Application Catalog.
	Not connected to an Application Catalog	Package Expert is not connected to an Application Catalog.

Output Window Toggle

As PackageExpert executes tests or performs resolutions, messages appear in the Output Window.

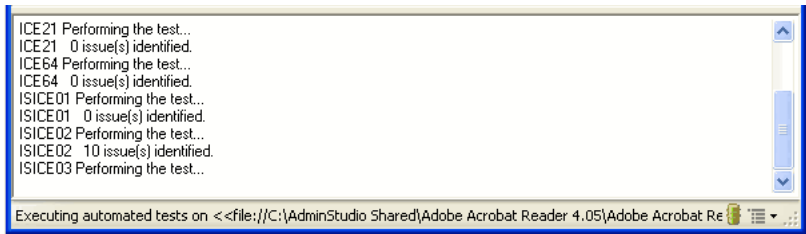

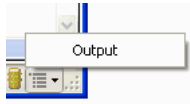


Figure 14-21: Test Execution Messages in the PackageExpert Output Window

You can toggle the display of the Output Window open or closed by clicking on the menu icon  in the Status Bar and selecting **Output** from the list.



PackageExpert Views

The PackageExpert interface includes the following views:

Table 14-12 • PackageExpert Views

View	Description
Test and Resolve View	Perform testing and view test results.
Configuration View	Specify the default set of PackageExpert tests to run during subsequent test runs.
Filter View	Edit or delete existing test result filters.
Options View	Specify the resolution method that you want PackageExpert to use.

Test and Resolve View

The **Test and Resolve** view displays a package tree containing packages and groups of packages from a selected Application Catalog. The Right Pane view varies depending upon what is selected in the package tree:

- [File System Node Selected](#)
- [Package Selected](#)
- [Group Selected](#)
- [Application Catalog Selected](#)
- [Test Category Selected](#)
- [Message Node Selected](#)

File System Node Selected

When the **File System** node is selected, the **Select a Package** form is displayed in the right pane, and prompts you to select a Windows Installer package to test. See [Select a Package Dialog Box](#) for more information.

Package Selected

When a package is selected in the package tree, the following information is displayed.

Table 14-13 • Test & Resolve View / Package Selected

Item	Description
Package Name	The name of the selected package is displayed in the banner at the top of the right pane.
Manufacturer	The manufacturer of the this Windows Installer package.
File	Location of the Windows Installer package that was opened.
Test Results for All Categories	Listing of all of the messages that were generated during the last test run.
Output Window	Displays messages produced during testing and resolution.

Group Selected

When a group is selected, summary information about the Test Results of the packages in that group are displayed:

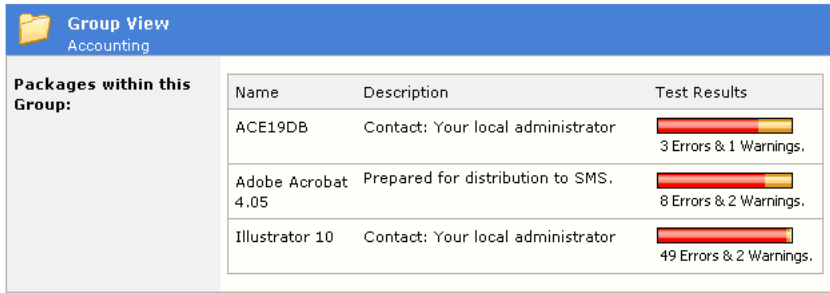


Figure 14-22: Group Summary Information

Application Catalog Selected

When the Application Catalog node is selected, and you are connected to an Application Catalog, summary information about the Test Results of the packages in the Application Catalog are displayed:

Application Catalog View

ABC04SMITH\XP\SQL\EXPRESS-sample

Statistics:

The Application Catalog contains:

- 8 MSI packages.

Most Recently Tested Packages:

Details	Date
ACE19DB	10/24/2006 7:27 PM
Illustrator 10	10/24/2006 7:27 PM
Adobe Acrobat 4.05	10/24/2006 6:00 PM

Percentage of Application Catalog Tested:

100% (8 out of 8 MSI Packages Tested).

6 package(s) with Errors & 6 package(s) with Warnings.

Error

Warning

Application Catalog History:

Details	User	Date
Database created.	sa	9/28/2006 7:37 AM

Figure 14-23: Application Catalog Summary Information

Test Category Selected

When a Test Category node is selected test results for all of the tests in that category are listed.

ACE19DB

Package View

Manufacturer:

Your Company Name

File:

C:\AdminStudio\Shared\ACE19\ACE19DB\ACE19DB.msi

Test Results for All Categories





	Name	Status	Description	Help
	Restart Manager Files In Use Dialog	Ran	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista or later.	Info
	ISICE02	Ran	The 'Notepad.exe' file in 'NewComponent1' component should be digitally signed for proper support with Windows Vista or later.	Info
	ISICE09	Ran	An MsiPatchCertificate table entry	Info

Figure 14-24: Test Category Summary Information

Message Node Selected


When a message node in the package tree is selected, detailed information on that specific message is displayed, including the exact text of the selected error message and a help topic that explains what the test is checking for and how errors are resolved.

 Microsoft_ProjectStd_2002
Restart Manager Files In Use Dialog

Manufacturer: Microsoft Corporation
File: C:\AdminStudio\Shared\PackageExpert\MicrosoftProjectStd2002.msi

Error Details

Item	Details
Test Name	Restart Manager Files In Use Dialog / Error Number 1
Description	The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista.
Status	Ran
Details	Checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.
Help	TestHelpTopics\Restart Manager Files In Use Dialog.htm

 **Restart Manager Files In Use Dialog**
Checks for Presence of MsiRMFilesInUse Dialog

Description

The Restart Manager Files In Use Dialog test checks to see if the MsiRMFilesInUse dialog is present in the Windows Installer (.msi) package.

Error Message

If this test detects that the MsiRMFilesInUse dialog is present in the package, the following message is displayed:

The 'Restart Manager Files In Use Dialog' test ran, but did not detect any problems as the 'MsiRMFilesInUse' dialog is present in the MSI.

Resolution: Automatic

PackageExpert automatically resolves this error by adding the MsiRMFilesInUse dialog to the package. In the output, the following message is listed:

The 'MsiRMFilesInUse' dialog was added to the MSI.

Figure 14-25: Message Node Detailed Information

Configuration View

On the **Configuration** view, you specify which tests you want to PackageExpert to run for all subsequent test runs. These settings can be changed at any time and any changes you make do not effect the results of packages that have already been tested.

The Configuration view consists of two panes:

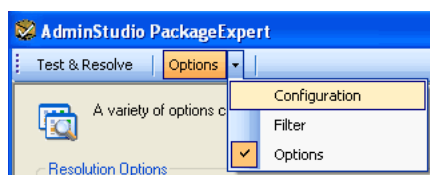
- **Tests tree**—Tests are grouped into categories and are displayed in a tree on the left pane. Select the tests that you want PackageExpert to run for subsequent test runs.



Tip • You can select or unselect all of the tests in a test category by selecting or unselecting the test category check box.

- **Help pane**—When you select a test, a help topic explaining that test opens in the right pane.

The **Configuration** view is opened by clicking **Configuration** in the PackageExpert toolbar menu. This toolbar menu toggles between **Configuration**, **Options**, and **Filter**. If **Configuration** is not displayed, click the arrow next to the menu item and select **Configuration** from the list.



Filter View

The **Filter** view lists all of the defined severity filters in the **Filter List**, all of the test result exclusion filters in the **Exclusion List**.

From the Filter view, you can **Edit** or **Delete** an existing filter.



Note • Filters are created by selecting a test result in the **Test & Resolve** view, and selecting **Filter Result** from the context menu. The **Test Result Filter Editor Dialog Box** opens, where you can create a filter. See [Customizing Test Results](#) for more information.

Editing a Result Filter

To edit an existing filter, select the filter and click the **Edit** button. The **Test Result Filter Editor** dialog box opens where you can edit the **Filter Severity** and/or **Description** fields and click **Update** to save your edits.

Deleting a Result Filter

To delete an existing filter, select the filter and click the **Delete** button.

Options View

On the **Options** view you can specify whether you want PackageExpert to resolve errors by creating a transform (.mst) file or by directly editing the .msi package.

You open the **Options** view by clicking the down arrow next to **Configuration** in the PackageExpert toolbar and selecting **Options**.

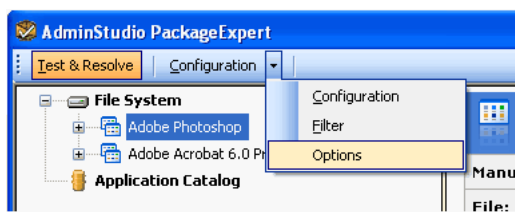


Figure 14-26: Opening the **Options** View

On the **Options** view, select one of the following options to specify the resolution method you want PackageExpert to use when automatically resolving errors:

- **Original MSI file**—Select this option to have PackageExpert directly edit the .msi file to resolve errors.
- **A newly created transform (MST) file**—Select this option to have PackageExpert create a transform (.mst) file to resolve errors.

PackageExpert Dialog Boxes

PackageExpert includes the following dialog boxes:

- [Connect Application Catalog Dialog Box](#)
- [Custom Test Results Editor Dialog Box](#)
- [Manual Test Perform Dialog Box](#)
- [Manual Test Resolve Dialog Box](#)
- [Select a Package Dialog Box](#)
- [Test Result Filter Editor Dialog Box](#)
- [Unpublished Packages Dialog Box](#)

Connect Application Catalog Dialog Box

On the **Connect Application Catalog** dialog box, which opens when you click on the Application Catalog node in the package tree, you specify the login information to connect to an Application Catalog database. This dialog box has two tabs:

- **Enterprise Server**—Select this tab to open the AdminStudio Enterprise Server Application Catalog database. See [Enterprise Server Tab](#).
- **Standalone**—Select this tab to open an Application Catalog database other than the AdminStudio Enterprise Server Application Catalog database. See [Standalone Tab](#).

Making this the Default Shared Application Catalog

If you select the **Make this the default shared Application Catalog** option, the Application Catalog you are opening will become the default Application Catalog (and be recorded as such in the **AdminStudio Shared** directory).

If the Application Catalog is made the default, all other AdminStudio users that use the same shared directory will automatically connect to the default Application Catalog when AdminStudio is launched. Therefore, you should only set this option if you want to affect all AdminStudio users who access that shared directory.



Note • In the AdminStudio Enterprise Edition, only the AdminStudio Administrator or users with the Change Default Database permission will see the **Make this the default shared Application Catalog** option. This allows the AdminStudio Administrator to configure the default Application Catalog, and then subsequent installations of AdminStudio will automatically connect to the default Application Catalog if they use the same shared directory.

Enterprise Server Tab

To connect to the AdminStudio Enterprise Server Application Catalog, you log in on the **Enterprise Server** tab of the **Connect Application Catalog** dialog box.

The **Enterprise Server** tab includes the following options:

Table 14-14 • Connect Application Catalog Dialog Box / Enterprise Server Tab Options

Option	Description
URL	<p>A link indicating the location of the AdminStudio Enterprise Server associated with this installation of AdminStudio.</p> <p>You can click on this link to open the Select AdminStudio Enterprise Server URL dialog box to identify the AdminStudio Enterprise Server URL that you would like to connect to.</p>
Authentication	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Windows Authentication • AdminStudio Enterprise Server User
User Name and Password	<p>If you chose AdminStudio Enterprise Server User, enter your AdminStudio Enterprise Server User Name and Password (provided by your System Administrator).</p>

Standalone Tab

On the **Standalone** tab of the **Connect Application Catalog** dialog box, enter the information required to login to the specified Application Catalog.

The **Standalone** tab includes the following options:

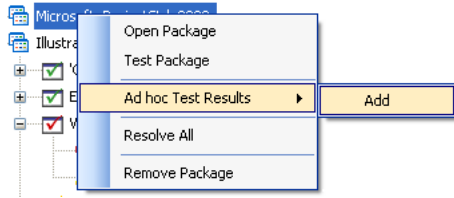
Table 14-15 • Connect Application Catalog Dialog Box / Standalone Tab Options

Option	Description
Server	Select one of the available SQL Servers on the network from this list. You can also manually enter the name of the SQL Server to which you want to connect.
Authentication	Select one of the following options: <ul style="list-style-type: none">• Windows Authentication—Choose to use Windows network authentication (your network login ID) to log into this Application Catalog.• Server Authentication—Choose to use SQL Server login identification for authentication.• Login ID and Password—If you chose Server Authentication, enter the appropriate Login ID and Password.
Catalog	This is a list of catalogs available on the specified server. Select the name of the Application Catalog database that you want to connect to.
Test	Click this button to test whether a connection can be made to the database.
Make this the default shared Application Catalog	<p>When this option is selected, the Application Catalog you are trying to open or create will become the default Application Catalog (and be recorded as such in the AdminStudio Shared directory).</p> <p>If the Application Catalog is made the default, all other AdminStudio users that use the same shared directory will automatically connect to the default Application Catalog when AdminStudio is launched. Therefore, you should only set this option if you want to affect all AdminStudio users who access that shared directory.</p>

Custom Test Results Editor Dialog Box

The **Custom Test Results Editor** dialog box is used to manually add a test result message to a package. You might want to do this if you tested the package using QualityMonitor, InstallShield, or another tool, and you noticed something which is unexpected and which you want to document. Also, if a package failed during installation, you might also want to document the error.


The **Custom Test Results Editor** dialog box is opened by selecting a node in the package tree, pointing to **Ad hoc Test Results** and clicking **Add** on the context menu.



For example, suppose an error was encountered during installation, which caused installation failure, and you want to document this error.

When adding a test result on the **Custom Test Results Editor** dialog box, enter the following information:

Table 14-16 • Custom Test Results Editor Dialog Box Options

Option	Description
Category	<p>Either enter a new name for a test category where you want to display this message, or select an existing category from the list.</p>  <p>Note • You can only associate an ad-hoc test result with a test category you created, not one that contains tests shipped with PackageExpert. Ad-hoc test results must always be stored in manually-created categories. If a category has already been added, then it will be listed in the Category list. You can create a new category by entering a category name in the Category box.</p>
Name	Either enter a new name for the test, or—if the same test has generated more than one error—select an existing name from the list
Severity	Select Error , Warning , or Information to characterize the severity of this error.
Description	Document the error that you discovered during ad-hoc testing. This information will be displayed under Error Details in the right pane when this test result node is selected.

The following dialog box displays an example of an added message:

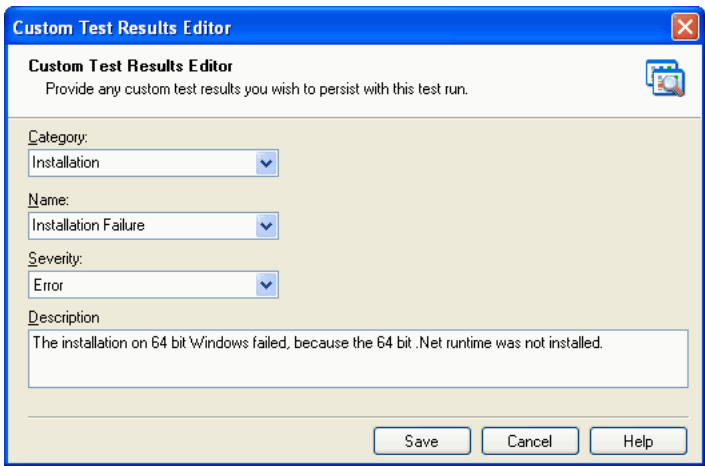


Figure 14-27: Example of a Message Added to Document Ad-Hoc Testing

When you click **Save** on the **Custom Test Results Editor** dialog box, the new message will be displayed in the package tree, and will be documented in the **Error Details** area in the right pane.

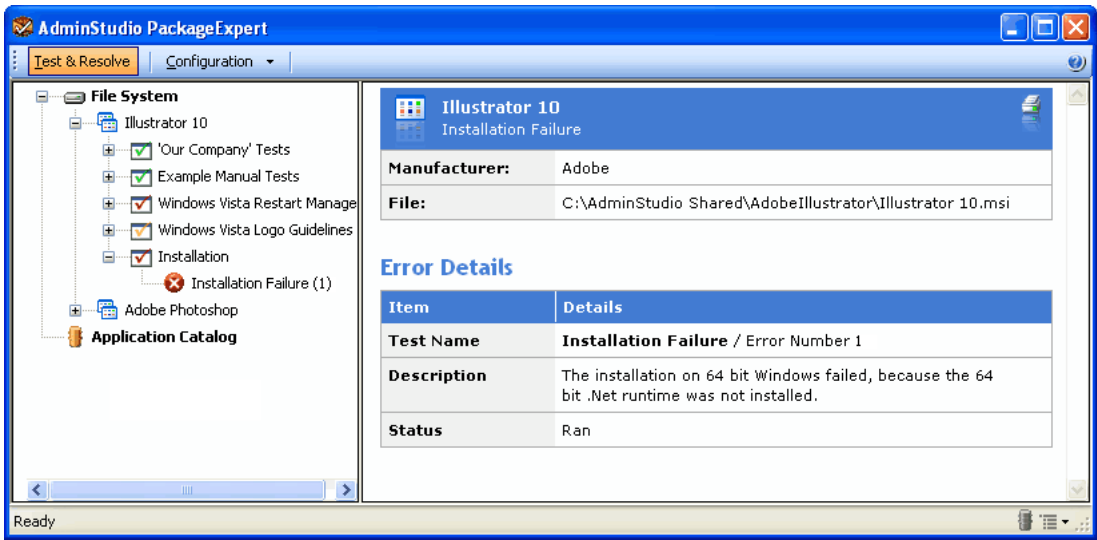


Figure 14-28: Display of a Message Added to Document Ad-Hoc Testing

If the future, when this problem has been corrected, you could delete or demote the severity of the test

Manual Test Perform Dialog Box

The **Manual Test Perform** dialog box opens when you choose to run a Manual Test that you have added to PackageExpert and contains instructions on how to perform the Manual Test.

The steps that are listed on the **Manual Test Perform** dialog box come from the XML file associated with this Manual Test. As you manually perform each listed step, you then right-click on the step and set its status to either **Pass** or **Fail**. See [Adding Manual Tests](#) for more information.

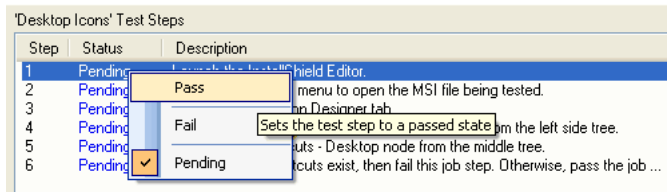


Figure 14-29: Changing the Status of a Step in a Manual Test

You open the **Manual Test Perform** dialog box by selecting a Manual Test in the package tree and selecting **Perform Manual Test** from the context menu.

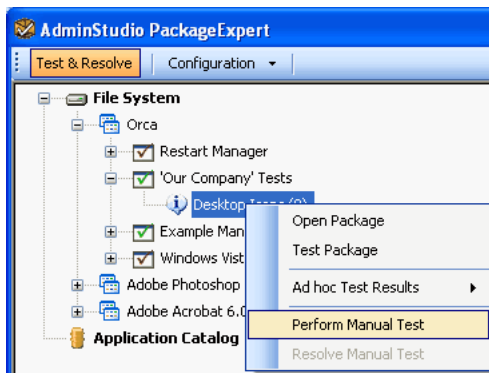


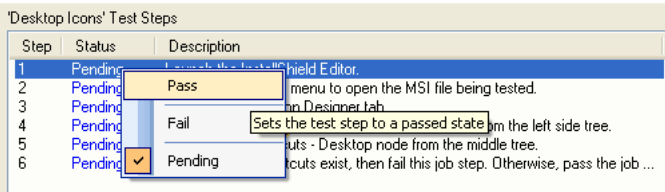
Figure 14-30: Opening the Manual Test Perform Dialog Box



Note • In order to perform a Manual test, the Test Results mode of the package must be set to **Local**.

The Manual Test Perform dialog box includes the following options:

Table 14-17 • Manual Test Perform Dialog Box

Option	Description
Step	Numbers the steps in this procedure.
Status	States whether the step has not yet been performed (Pending), whether it was run without errors (Pass), or whether was run and produced errors (Fail). Right-click on the listed status to change it: 
Description	Contains the instructions for performing this Manual Test.
Save	Click to save the status changes that you have made.

Manual Test Resolve Dialog Box

For Manual Tests that are added to PackageExpert, you use the **Manual Test Resolve** dialog box to document the completion of steps taken to manually resolve a test result. The **Manual Test Resolve** dialog box lists the manual steps you need to perform to resolve this error.

You open the **Manual Test Resolve** dialog box by selecting a Manual Test in the package tree, and selecting **Resolve Manual Test** from the context menu.

The steps that are listed on the **Manual Test Resolve** dialog box come from the XML file associated with this Manual Test.

The first time you open this dialog box, all of the steps are marked **Incomplete**. Perform each of the listed steps, marking each as **Complete** after you have performed them, and then click **Save**.

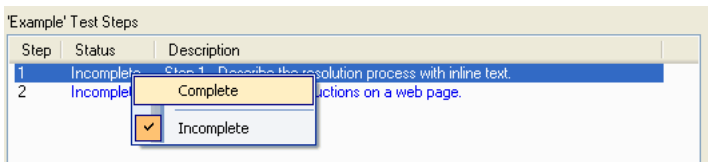
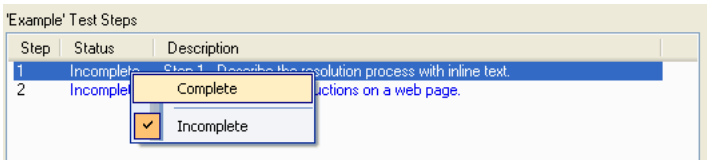


Figure 14-31: Marking Steps Complete When Resolving a Manual Test

You open the **Manual Test Resolve** dialog box by selecting a Manual Test in the package tree and selecting **Resolve Manual Test** from the context menu.

The Manual Test Resolve dialog box includes the following options:

Table 14-18 • Manual Test Resolve Dialog Box

Option	Description
Step	Numbers the steps in this resolution procedure.
Status	<p>States whether the step has been Completed or is still Incomplete.</p> <p>Right-click on the listed status to change it:</p> 
Description	Contains the instructions for performing this resolution procedure.
Save	Click to save the status changes that you have made.

Select a Package Dialog Box

If you select the **File System** node in the package tree, and select **Open Package** on the context menu, the Select a Package dialog box opens, prompting you to browse to a Windows Installer package on your file system.

The Select a Package dialog box includes the following options:

Table 14-19 • Select a Package Dialog Box Options

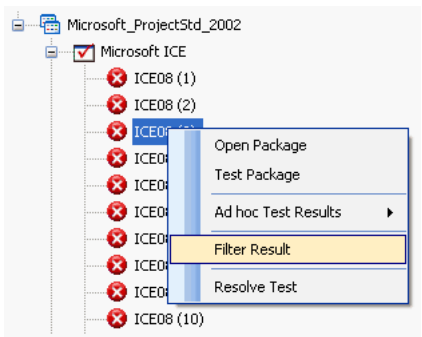
Option	Description
MSI Package	Click Browse and select a Windows Installer package on your local file system that you want to test.
Transform List	<p>After you have selected an MSI Package, if any transform files exist in the same directory as the .msi file, they are automatically listed in the Transform List, but are not selected.</p> <ul style="list-style-type: none"> • To include listed transform files—Select the transform files that you want to include in PackageExpert testing. • To add additional transform files—Click Browse and select a transform file to add it to the Transform list, and then select it to include it in testing.
Test	Click to open the selected package and perform testing on the package.
Cancel	Click to open the selected package without performing testing.

Test Result Filter Editor Dialog Box

You can use the **Test Result Filter Editor** dialog box to customize PackageExpert test results to accommodate unique or special circumstances at your organization. You can select a test result and change its severity or exclude it from the listings. You make these changes by making a selection from the **Filter Severity** list:

- **Change severity**—To change the severity of a specific test result from its original severity, select **Error**, **Warning**, or **Information**.
- **Exclude**—To suppress the display of a specific test result, select **Exclude**.

You open the Test Result Filter Editor dialog box by opening the **Test & Resolve** view, selecting the specific message that you want to modify and selecting **Filter Result** from the context menu.



Note • You are filtering a specific instance of a message generated by a test, not all messages generated by that test. For example, suppose the ICE08 test produced five error messages. You consider messages 1 through 4 to be valid errors. However, message 5 concerns a particular component of the package that no one at your organization is likely to use, and therefore you do not want this particular test result to be displayed. You could then change the Severity to **Exclude** to suppress the display of this test result.

The **Test Result Filter Editor** dialog box includes the following options:

Table 14-20 • Test Result Filter Editor Dialog Box

Option	Description
Test Result Area	<p>Lists the Name and Category of the test that produced this message, and lists the message text of the test Result you selected.</p> <p>For example, this is a message produced by the Restart Manager Files in Use Dialog test :</p> <p>The 'MsiRMFilesInUse' dialog is needed to support the Restart Manager on Windows Vista or later.</p>

Table 14-20 • Test Result Filter Editor Dialog Box

Option	Description
Filter Severity	<p>Select one of the following options to filter this message:</p> <ul style="list-style-type: none"> • Error—Select to identify this test result as a problem that <i>would</i> cause incorrect behavior when this package is installed or used. • Warning—Select to identify this test result as a problem that <i>could</i> cause incorrect behavior when this package is installed or used. • Information—Select to indicate that this test result is not an error. • Exclude—Select to exclude the display of this test result. <p>After you have set this filter, if the same problem is encountered in another package or in a subsequent test run of this package, the filter that you defined will continue to be used.</p>
Description	Enter an explanation for why you changed the severity level or excluded this message
Save	Click to save the filter.

Unpublished Packages Dialog Box

This dialog box opens if you attempt to exit PackageExpert before you have published test results to the Application Catalog for those packages whose **Test Results** mode is set to **Catalog**.

Select those packages in the **Packages** list for which you want to save test results to the Application Catalog, and click **OK**.

Analyzing the Impact of Installing Microsoft Operating System Patches



Edition • OS Security Patch Wizard is included with AdminStudio Enterprise Edition.

OS Security Patch Wizard is a new AdminStudio tool that is used to import Microsoft operating system patch information into the Application Catalog.

You need to import Microsoft OS patch information into the Application Catalog so that you can analyze the full impact of installing these patches on user machines. Based on the analysis results, you can determine the level of testing you need to perform before distributing a Microsoft OS patch throughout your enterprise.

Information about the OS Security Patch Wizard and patch impact analysis is presented in the following sections:

Table 15-1 • OS Security Patch Wizard/Patch Impact Analysis Wizard Help Library

Section	Description
About Microsoft Operating System Patches	Explains what Microsoft operating system patches are, and why you should include them in your package testing processes.
Using OS Security Patch Wizard to Import Microsoft OS Patches	Explains how to download and import a Microsoft operating system patch into the Application Catalog. <ul style="list-style-type: none">• Identifying and Downloading Microsoft Operating System Patch Files• Importing a Microsoft Operating System Security Patch Into the Application Catalog
Analyzing the Impact of Installing a Microsoft Operating System Patch	Explains how to use the ConflictSolver Patch Impact Analysis Wizard to identify conflicts between Microsoft operating system patches and the packages and OS Snapshots in the Application Catalog.
Reference	Describes all of the panels in the OS Security Patch Wizard.

About Microsoft Operating System Patches

Each month, Microsoft releases patches to address security vulnerabilities that are discovered in Microsoft operating system software. Microsoft defines security vulnerabilities as:

"A flaw in a product that makes it infeasible – even when using the product properly—to prevent an attacker from usurping privileges on the user's system, regulating its operation, compromising data on it, or assuming ungranted trust."

Microsoft publishes a monthly Microsoft Security Bulletin Summary that lists the patches released that month. You can view these bulletins on the [Microsoft TechNet Security Center](http://www.microsoft.com/technet/security/bulletin/ms07-sep.msp) Web site. The Microsoft Security Bulletin Summary for September 2007 is shown in the following figure:



Figure 15-1: Microsoft Security Bulletin Summary for September 2007

The Security Bulletin Summary lists each patch released that month, grouped by status level, with a link to each patch's associated Security Bulletin, as shown in the following figure:

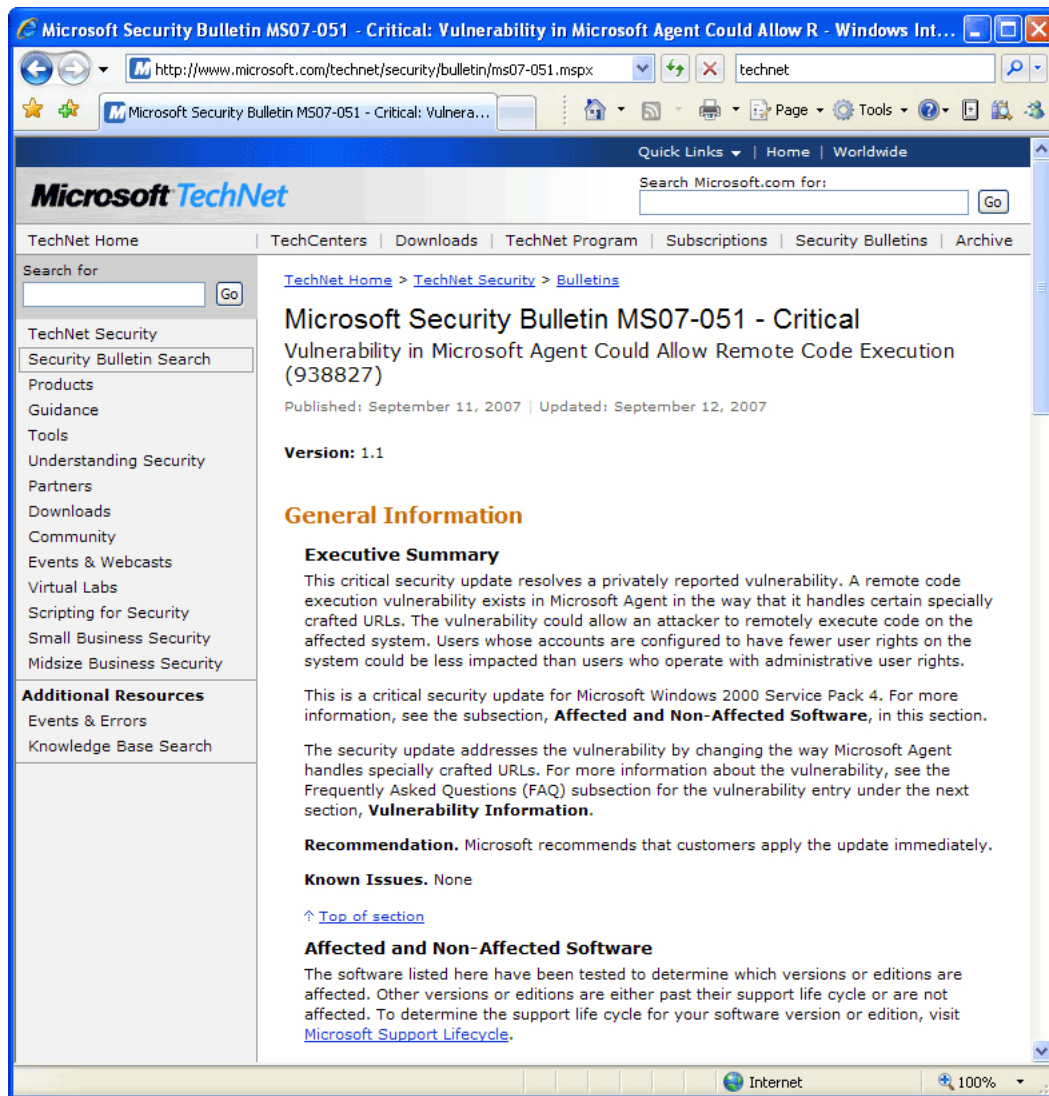


Figure 15-2: Microsoft Security Bulletin MS07-051

Security Bulletin Summaries, Security Bulletins, and patches can be accessed from the [Microsoft TechNet Security Center](http://www.microsoft.com/technet/security/default.mspx):

<http://www.microsoft.com/technet/security/default.mspx>

Using OS Security Patch Wizard to Import Microsoft OS Patches

You can use the OS Security Patch Wizard to import Microsoft operating system patch information into the AdminStudio Application Catalog. When you select the patch to import, you are prompted to review the imported data and to add information from the patch's Security Bulletin. Information about importing Microsoft operating system patches is presented in the following sections:

- [Identifying and Downloading Microsoft Operating System Patch Files](#)
- [Importing a Microsoft Operating System Security Patch Into the Application Catalog](#)

Identifying and Downloading Microsoft Operating System Patch Files

To identify and obtain the Microsoft operating system patch files that you want to import into the Application Catalog, perform the following steps.



Task: *To identify and download Microsoft OS patch files:*

1. Open the Microsoft Security Bulletin Summary that lists the patch that you want to import. The following figure is of the Microsoft Security Bulletin Summary for September 2007 :

[TechNet Home](#) > [TechNet Security](#) > [Bulletins](#)

Microsoft Security Bulletin Summary for September 2007

Published: September 11, 2007 | Updated: September 12, 2007

Version: 1.1

This bulletin summary lists security bulletins released for September 2007.

With the release of the bulletins for September 2007, this bulletin summary replaces the bulletin advance notification originally issued September 6, 2007. For more information about the bulletin advance notification service, see [Microsoft Security Bulletin Advance Notification](#).

For information about how to receive automatic notifications whenever Microsoft security bulletins are issued, visit [Microsoft Technical Security Notifications](#).

Microsoft is hosting a webcast to address customer questions on these bulletins on September 12, 2007, at 11:00 AM Pacific Time (US & Canada). [Register now for the September Security Bulletin Webcast](#). After this date, this webcast is available on-demand. For more information, see [Microsoft Security Bulletin Summaries and Webcasts](#).

Microsoft also provides information to help customers prioritize monthly security updates with any non-security, high-priority updates that are being released on the same day as the monthly security updates. Please see the section, **Other Information**.

Bulletin Information

Executive Summaries

The security bulletins for this month are as follows, in order of severity:

- ▣ **Critical (1)**
- ▣ **Important (3)**

[↑ Top of section](#)

- ▣ **Affected Software and Download Locations**
- ▣ **Detection and Deployment Tools and Guidance**

In this example, three patches are listed under **Executive Summaries** in the **Bulletin Information** section: one with a status of **Critical** and two with a status of **Important**.

- Expand the listing of the patch status that you want to review. In this example, if you expanded the **Critical** status, the following **Executive Summary** of Microsoft Security Bulletin MS07-051 is displayed:

Bulletin Information

Executive Summaries

The security bulletins for this month are as follows, in order of severity:

❏ **Critical (1)**

Bulletin Identifier	Microsoft Security Bulletin MS07-051
Bulletin Title	Vulnerability in Microsoft Agent Could Allow Remote Code Execution (938827)
Executive Summary	This critical security update resolves a privately reported vulnerability. A remote code execution vulnerability exists in Microsoft Agent in the way that it handles certain specially crafted URLs. The vulnerability could allow an attacker to remotely execute code on the affected system. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.
Maximum Severity Rating	Critical
Impact of Vulnerability	Remote Code Execution
Detection	Microsoft Baseline Security Analyzer can detect whether your computer system requires this update. The update will require a restart.
Affected Software	Windows. For more information, see the Affected Software and Download Locations section.

- Click the link in the **Bulletin Title** field. The Security Bulletin for that patch opens. In this example, Security Bulletin MS07-051 opens:

Microsoft Security Bulletin MS07-051 - Critical

Vulnerability in Microsoft Agent Could Allow Remote Code Execution (938827)

Published: September 11, 2007 | Updated: September 12, 2007

Version: 1.1

General Information

Executive Summary

This critical security update resolves a privately reported vulnerability. A remote code execution vulnerability exists in Microsoft Agent in the way that it handles certain specially crafted URLs. The vulnerability could allow an attacker to remotely execute code on the affected system. Users whose accounts are configured to have fewer user rights on the system could be less impacted than users who operate with administrative user rights.

This is a critical security update for Microsoft Windows 2000 Service Pack 4. For more information, see the subsection, **Affected and Non-Affected Software**, in this section.

The security update addresses the vulnerability by changing the way Microsoft Agent handles specially crafted URLs. For more information about the vulnerability, see the Frequently Asked Questions (FAQ) subsection for the vulnerability entry under the next section, **Vulnerability Information**.

Recommendation. Microsoft recommends that customers apply the update immediately.

Known Issues. None

[↑ Top of section](#)

Affected and Non-Affected Software

The software listed here have been tested to determine which versions or editions are affected. Other versions or editions are either past their support life cycle or are not affected. To determine the support life cycle for your software version or edition, visit [Microsoft Support Lifecycle](#).

Chapter 15: Analyzing the Impact of Installing Microsoft Operating System Patches

Using OS Security Patch Wizard to Import Microsoft OS Patches

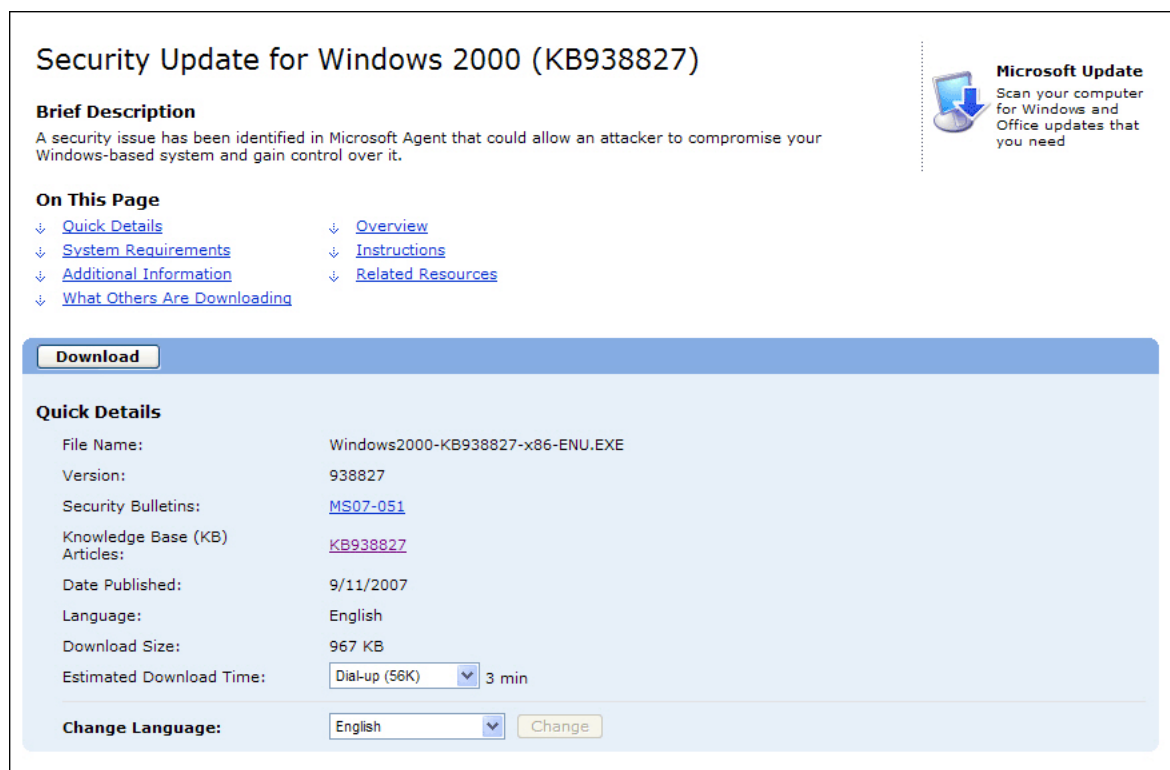
In the Security Bulletin, the **Affected Software** table lists the software affected by this patch, and provides a link to the download page for that specific patch:

Affected Software			
Operating System	Maximum Security Impact	Aggregate Severity Rating	Bulletins Replaced by This Update
Microsoft Windows 2000 Service Pack 4	Remote Code Execution	Critical	MS07-020

For some patches, both an **Operating System** and **Component** are listed, while for others, only an **Operating System** is listed.

Affected Software				
Operating System	Component	Maximum Security Impact	Aggregate Severity Rating	Bulletins Replaced by This Update
Windows 2000 Service Pack 4	Windows Services for UNIX 3.0	Elevation of Privilege	Important	None
Windows 2000 Service Pack 4	Windows Services for UNIX 3.5	Elevation of Privilege	Important	None
Windows XP Service Pack 2	Windows Services for UNIX 3.0	Elevation of Privilege	Important	None
Windows XP Service Pack 2	Windows Services for UNIX 3.5	Elevation of Privilege	Important	None
Windows Server 2003 Service Pack 1 and Windows Server 2003 Service Pack 2	Windows Services for UNIX 3.0	Elevation of Privilege	Important	None
Windows Server 2003 Service Pack 1 and Windows Server 2003 Service Pack 2	Windows Services for UNIX 3.5	Elevation of Privilege	Important	None
Windows Server 2003 Service Pack 1 and Windows Server 2003 Service Pack 2	Subsystem for UNIX-based Applications	Elevation of Privilege	Important	None
Windows Server 2003 x64 Edition and Windows Server 2003 x64 Edition Service Pack 2	Subsystem for UNIX-based Applications	Elevation of Privilege	Important	None
Windows Vista	Subsystem for UNIX-based Applications	Elevation of Privilege	Important	None
Windows Vista x64 Edition	Subsystem for UNIX-based Applications	Elevation of Privilege	Important	None

4. In the **Affected Software** table, click the link of the patch you want to import. The download page for that patch opens.



5. Click the **Download** button to download the patch. After the download is complete, the **Thank You for Downloading** page opens.

Importing a Microsoft Operating System Security Patch Into the Application Catalog

To import a Microsoft operating system security patch file into the Application Catalog, perform the following steps:



Task: *To use OS Security Patch Wizard to import a Microsoft OS patch file into the Application Catalog:*

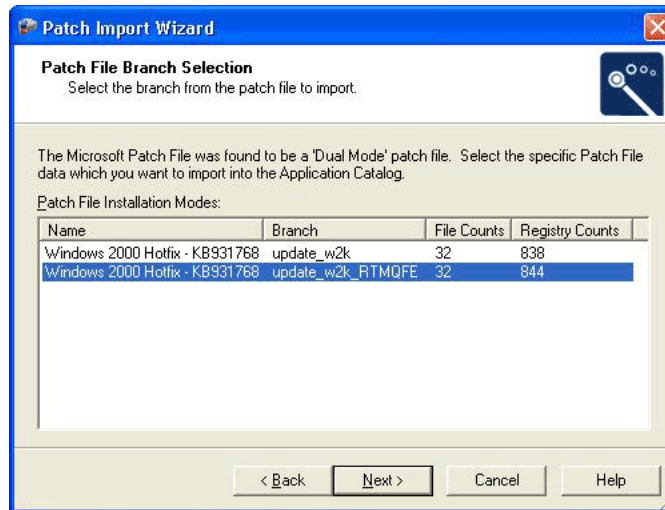
1. Open **OS Security Patch Wizard**. The **Welcome** panel opens.
2. Click **Next**. The **Patch File Selection** panel opens.
3. Click **Browse** and select a Microsoft patch file that you have downloaded from the [Microsoft TechNet Security Center](#), as described in [Identifying and Downloading Microsoft Operating System Patch Files](#).

4. Click **Next**. OS Security Patch Wizard reads and verifies the patch file and displays progress messages in the **Processing Status** area of this panel.
 - **Prompt to connect to the Application Catalog**—If you opened OS Security Patch Wizard from the **Start** menu, after you click **Next**, you will be prompted to provide database connection information. If you opened OS Security Patch Wizard from Application Manager, you will already be connected to the Application Catalog and will not be prompted to connect again.
 - **Invalid patch file message**—If you are attempting to import a patch file that is invalid, a message will be displayed and you will be unable to proceed with the import.

When processing of the patch file is successfully finished, one of the following panels opens:

- If you have downloaded a [dual-mode patch](#), the **Patch File Branch Selection** panel opens. Continue with [Step 5](#) below
 - If the patch you are importing is not a dual-mode patch, the **Patch Bulletin Information** panel opens. Skip to [Step 7](#) below.
5. Windows XP software updates that are released after Service Pack 1 (SP1) may include binary files for both Windows XP-based and Windows XP-based SP1 computers. Therefore, there are multiple processing paths for installing this patch. These are called *dual-mode* patches. If you are importing a dual-mode patch, the **Patch File Branch Selection** panel opens.

On this panel, you need to identify the branch of the patch file that you want to import into the Application Catalog. AdminStudio has preselected the branch of this patch that appears to install the latest version. You may choose to accept the default selection or make your own selection.



Note • It is permissible to import the same dual-mode patch multiple times, selecting different branches each time. Subsequent panels will allow you to uniquely identify each patch branch. For more information on dual-mode patches and identifying their branches, see [About Dual-Mode OS Patches](#).

6. Click **Next**. The **Patch Bulletin Information** panel opens.

7. From the patch's associated Security Bulletin, copy and paste the **Bulletin ID** and the **Bulletin Title** associated with the patch file you are importing. You should also enter the **Bulletin Summary** and **Bulletin Release Date**.



Important • Security Bulletin data is used by Application Manager to integrate with the Microsoft Web site. Therefore, it is important that you accurately enter this data.



Note • For information on how to view Microsoft Security Bulletins, see [Identifying and Downloading Microsoft Operating System Patch Files](#).

8. Click **Next**. The **Additional Patch Information** panel opens, displaying some data that was extracted from the patch file.
9. Review and edit the following information:

- **Name**—By default, this field displays the name of the selected patch file, such as:

Windows2000-KB38827-x86-ENU.EXE

You should edit this name to make it easier for you to identify this patch when it is listed in the Application Catalog.

- **Associated KB Article**—The KB article number is extracted from the patch file and automatically entered in this field.



Important • The Knowledge Base article identification information is used by Application Manager to integrate with the Microsoft Web site. Therefore, it is important that you accurately enter this data.

- **Effected Product**—Optionally, enter the name of the product that this patch effects. This information is typically something like Windows XP, Windows Vista, etc.
- **Additional Comments**—Comments are extracted from the patch file and are automatically entered in this field. You can edit these comments.

10. Click **Next**. The **Patch Import Summary** panel opens, listing the selections you have made.
11. Click **Import** to initiate the patch import. When import is complete, the **Import Process Results** panel opens, listing the results of the import of the selected patch file. The results displayed on this panel, along with the summary information shown in the previous panel, are output to a log file.



Note • When you click **Import**, if you have not previously connected to the Application Catalog, you will be prompted to provide database connection information.

12. To import another patch, click **< Another** to return to the beginning of the OS Security Patch Wizard.
13. When you have finished importing patches, click **Finish**.

Information About Missing Patch Files

The OS Security Patch Wizard imports a Microsoft operating system patch using the patch's INF file. An INF file is a text file that is used by Microsoft to install the patch and which contains the information necessary to install a patch, such as for a driver, including file names and locations, registry information, and version information.

An OS Security Patch can consolidate patches for different Windows Operating Systems. Any given INF file may contain information about files that do not exist or are not relevant for a particular Windows Operating System. The OS Security Patch Wizard ignores these files when importing the patch information into the Application Catalog and reports the missing files in the log file. The OS Security Patch Wizard will not import any information about the missing files into the Application Catalog.

About Dual-Mode OS Patches

The OS Security Patch Wizard imports a Microsoft operating system patch using the patch's INF file, which is a text file that contains all the information necessary to install a device (such as INI and registry changes, filenames, and locations of the source files on source media).

The **Branch** listed in this column is the name of the patch's INF file minus its extension. Because all obvious identifiable information within the INF file (like **Name**) is often identical, this Branch information is included to help differentiate the patches within a dual mode patch.

A sample branch name could be: update_SP2GDR or update_SP2QFE, where:

- **QFE** = Quality Fix Engineering updates (also known as hot fixes). These include security updates, feature packs, update roll-ups, drivers, and critical updates. They are created when an issue is identified that needs a fix in a specific and small subset of customer environments. These fixes have not been tested for the same length of time for regressions or incompatibilities as what a GDR fix typically goes through.
- **GDR** = General Distribution Release updates. These updates can be found on the Windows Update site, meaning that they are available to a wider audience. GDRs are tested to a greater degree than QFEs.



Note • QFEs generally contain all the prior fixes from both GDR and QFE releases while GDR releases do not contain QFE fixes.

Analyzing the Impact of Installing a Microsoft Operating System Patch

After you have imported a Microsoft operating system patch into the AdminStudio Application Catalog, you can continue to use the ConflictSolver Patch Impact Analysis Wizard to identify conflicts between Microsoft operating system patches and the packages and OS Snapshots in the Application Catalog. This helps you determine how specific MSI packages or OS Snapshots would be affected when a Microsoft operating system patch is installed.



Important • The patch information that you import using OS Security Patch Wizard is identical to and compatible with the patch information that you imported in previous releases using the Patch Impact Manager tool. Therefore, any procedures you previously used to analyze patch information will continue to work as they did in previous releases.

The section is organized in the following topics:

- [Performing Patch Impact Analysis](#)
- [Viewing Patch Impact Analysis Results](#)
- [Viewing Patch and Patch Impact Information in Application Manager and ConflictSolver](#)
- [Generating the Patch Report](#)

Performing Patch Impact Analysis

You can use the Patch Impact Analysis Wizard to identify conflicts between Microsoft operating system patches and the packages and OS Snapshots in your Application Catalog. This helps you determine how specific MSI packages or OS Snapshots would be affected when a Microsoft OS patch is installed.



Task:

To perform patch impact analysis:

1. On the ConflictSolver **Patches** tab, select **Patch Impact Analysis Wizard** from the **Conflicts** menu or select a patch and select **Patch Impact Analysis Wizard** from the context menu.
 - If you had a patch selected when you launched the Wizard, the **Source Patches Panel** opens, with the imported patch(es) that were selected when you launched the wizard automatically selected.
 - If you launched the Wizard by selecting **Patch Impact Analysis Wizard** from the ConflictSolver **Conflicts** menu, the **Welcome Panel** will open.
2. Click **Next**. The **OS Snapshot Panel** opens.
3. Optionally, select an OS Snapshot to include in your analysis. The selected OS Snapshot will be used to identify specific file information for any patch impacts that are discovered. Only one OS Snapshot can be selected.



Tip • The OS Snapshot serves as a representation of the underlying baseline system in your enterprise. If you include an OS Snapshot in your Patch Impact Analysis, then the file version information displayed in warning messages is derived from that OS Snapshot. If an OS Snapshot is not included in the analysis, the file version information will be displayed as “unknown”. As such, it is recommended to include an OS Snapshot in your analysis because this allows a finer-tuned evaluation of impacts based upon file and version information.

4. Click **Next**. The **Source Patches Panel** opens.

5. On the **Source Patches Panel**, select the patches that you want to include in your analysis.

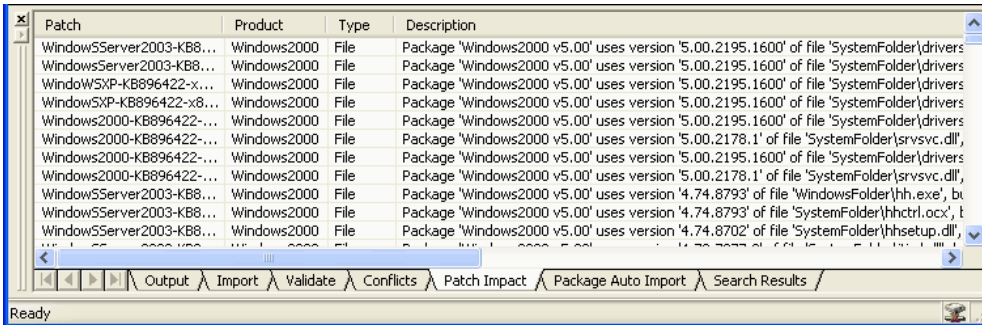
When searching for patches to include, you can use the **Filter by product** list to restrict the patches displayed on this panel. Also, to view more information on a patch, select the patch and click the **Patch Properties** button to open the **Patch Properties** dialog box.

6. After you have selected the patches that you want to include in your analysis, click **Next**. The **Target Products Panel** opens.
7. On the **Target Products Panel**, select the products or groups of products that you want to perform patch impact analysis on. The Patch Impact Analysis Wizard will analyze the products you select here against the patches you selected on the **Source Patches Panel** for impacts. You can use the **Select All** or **Clear All** buttons to make your selections.
8. Click **Next**. The **Summary Information Panel** opens, listing a summary of the options you selected in the Patch Impact Analysis Wizard.
9. Click **Finish** to accept these options and begin the Patch Impact Analysis. Analysis messages are listed in the **Output** tab of the ConflictSolver Output Window.

When analysis is complete, patch conflicts are listed on the **Patch Impact** tab of the Output Window in table format.

Viewing Patch Impact Analysis Results

After [Performing Patch Impact Analysis](#), patch conflicts are listed on the **Patch Impact** tab of the Output Window in table format.



Patch	Product	Type	Description
WindowsServer2003-KB8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
WindowsServer2003-KB8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
WindowsXP-KB896422-x8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
WindowsXP-KB896422-x8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
Windows2000-KB896422-...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
Windows2000-KB896422-...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2178.1' of file 'SystemFolder\srvcsv.dll'
Windows2000-KB896422-...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2195.1600' of file 'SystemFolder\drivers
Windows2000-KB896422-...	Windows2000	File	Package 'Windows2000 v5.00' uses version '5.00.2178.1' of file 'SystemFolder\srvcsv.dll'
WindowsServer2003-KB8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '4.74.8793' of file 'WindowsFolder\hh.exe', b
WindowsServer2003-KB8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '4.74.8793' of file 'SystemFolder\hhctrl.ocx', t
WindowsServer2003-KB8...	Windows2000	File	Package 'Windows2000 v5.00' uses version '4.74.8702' of file 'SystemFolder\hhsetup.dll'


Figure 15-3: Sample Patch Impact Analysis Results

The following information is included in Patch Impact Analysis results:

Table 15-2 • Information Included in Patch Impact Analysis Results

Item	Description
Patch	Name of a Windows Installer Patch.
Product	Name of a Package or an OS Snapshot in the Application Catalog.

Table 15-2 • Information Included in Patch Impact Analysis Results

Item	Description
Type	<p>Identifies the type of the impact as either a File or a Registry impact.</p>  <p>Note • Windows Installer Patches rarely impact Registry Entries, so most of the identified impacts will be identified as File.</p>
Description	<p>Description of how the Windows Installer Patch impacted with the package or OS Snapshot. For example:</p> <p>File 'CdrGfx.dll' in Package 'Coreldraw 12.0.0.458 v1.0' uses version 'Unknown' of file 'SystemFolder\RPCRT4.dll', but Patch 'WindowsXP-KB828741-x86-ENU.EXE' uses version '5.1.2600.1361' of the same file.</p> <p>This warning message means that:</p> <ul style="list-style-type: none"> • The Windows Installer Patch, WindowsXP-KB828741-x86-ENU.EXE, installs version 5.1.2600.1361 of RPCRT4.dll, an operating system file. <p>and</p> <ul style="list-style-type: none"> • Corel Draw, Coreldraw 12.0.0.458 v1.0, installs a file that is dependent upon the same operating system file: RPCRT4.dll. <p>Therefore, this warning message means that you should evaluate Corel Draw on a system that includes this Windows Installer Patch to insure that you can safely distribute this package within your enterprise.</p>  <p>Tip • The OS Snapshot serves as a representation of the underlying baseline system in your enterprise. If you include an OS Snapshot in your Patch Impact Analysis, then the file version information displayed in warning messages is derived from that OS Snapshot. If an OS Snapshot is not included in the analysis, the file version information will be displayed as “unknown”. As such, it is recommended to include an OS Snapshot in your analysis because this allows a finer-tuned evaluation of impacts based upon file and version information.</p>

Viewing Patch and Patch Impact Information in Application Manager and ConflictSolver

Patch content and analysis information can be viewed on the ConflictSolver and Application Manager Products and Patches tabs.

- [Viewing Patch Content Information](#)
- [Viewing Dependencies](#)
- [Viewing Associated Patches](#)
- [Viewing Patch Impacts on the Products Tab](#)
- [Viewing Patch Impacts on the Patches Tab](#)

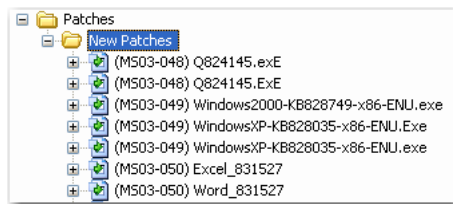
Viewing Patch Content Information



Task:

To view patch content information:

1. Launch ConflictSolver or Application Manager and click the **Patches** tab. The **All Patches View** opens.
2. Expand the listing. All of the patches that have been imported into the Application Catalog are listed. Newly imported patches are listed in the **New Patches** group.



In Application Manager, you can organize your patches into groups according to your business needs. See [Organizing Your Application Catalog Using Groups](#).

3. Select a patch. The **Patches View** opens.

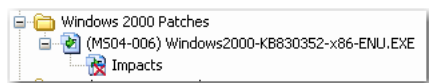
The following information on the selected patch is listed:

- **ID**—Microsoft Security Bulletin ID. Click on the ID number link to view this bulletin on the Microsoft Web site.
- **Title**—Title of patch
- **Release Date**—Date the patch was released by Microsoft.
- **KB Article**—Microsoft Knowledge Base article ID. Click on the KB Article link to view that article on the Microsoft Web site.
- **Imported On**—Date patch was imported into the Application Catalog

Chapter 15: Analyzing the Impact of Installing Microsoft Operating System Patches

Analyzing the Impact of Installing a Microsoft Operating System Patch

- **Groups**—Listing of which groups this patch belongs to. A patch can belong to multiple groups. You can copy a patch into multiple groups.
 - **Description**—You can enter a description of the patch in this field.
4. If an impact analysis was performed on a package using this patch, a **Patch Impacts** node is listed below the patch:



If you select the **Impacts** node, the **Impacts View** opens. See [Viewing Patch Impacts on the Patches Tab](#) for detailed information.

5. In Application Manager, you can view additional detailed patch information by selecting a patch and then selecting **Patch Properties** from the context menu. The **Patch Properties** dialog box opens.

The following information is listed:

- **General Tab**—View the title and a summary of a selected patch.
- **Contents Tab**—Lists all of the files and registry data contained in the selected patch.
- **Product Tab**—Lists the products that are updated by the selected patch.

Viewing Dependencies

On the ConflictSolver Dependencies View, which is accessed by selecting the **Dependencies** node under a product in the Product View, you can view a list of all of the files of a selected product that have dependencies with files used by other products or operating systems in the Application Catalog. This view is displayed for Windows Installer .msi packages in which file dependency information exists.

Dependency information is generated by the Patch Impact Analysis Wizard. You can also generate this information by selecting a product and then selecting **Scan for Dependencies** from the context menu.



Task:

To view product dependencies:

1. Launch ConflictSolver and click the **Products** tab. The **Group View** opens.
2. Select the product that you want to examine. The **Product View** for that product opens.
3. Expand the product to view its constituent views. If the **Dependencies** node appears, select it to display the [Dependencies View](#).

If the **Dependencies** node does not exist, select **Scan for Dependencies** from the context menu, and then select the **Dependencies** node.
4. Make selections from the **Files With Dependencies** list to further refine this listing, or select **(All)** to display all dependencies.
5. If you double-click on a dependency, the conflicting file on the target product will be displayed and highlighted in the Tables View.



Note • If the **Only Display View Nodes With Data** option on the **General** tab of the **ConflictSolver Options** dialog box is selected, if you scan for dependencies and no dependencies are found, the **Dependencies** node will still not be displayed.

Viewing Associated Patches

On the [Associated Patches View](#), you can view a list of imported patches that, if installed, would update that product.



Task:

To view associated patches:

1. Launch **ConflictSolver** and click the **Products** tab. The **Group View** opens.
2. Select the product that you want to examine. The **Product View** for that product opens.
3. Expand the product to view its constituent views and select the **Associated Patches** node. The [Associated Patches View](#) opens, displaying patches associated with that product.
4. If you double-click on a patch in the **Associated Patches View**, the Patch View (on the **Patches** tab) for that patch opens, listing general information on the selected patch.

See [Viewing Patch Content Information](#) for more information.

Viewing Patch Impacts on the Products Tab

The information listed on the [Patch Impact View \(Products Tab\)](#) depends upon the selection that is made in the **Impact category** list:

- When **Summary** is selected, the patches for which there is patch impact data persisted against the product are listed
- When **File Impacts** is selected, all impacts against this product are listed, and the patch that caused the impact is identified.



Task:

To view patch impacts:

1. Launch **ConflictSolver** and click the **Products** tab. The **Group View** opens.
2. Select the product that you want to examine. The **Product View** for that product opens.
3. Expand the product to view its constituent views and select the **Patch Impacts** node. The **Patch Impacts View** opens.
4. Select **Summary** from the **Impact category** list to view a list of patches for which there is patch impact data persisted against the product.
5. Select **File Impacts** from the **Impact category** list to view a list of all impacts against this product and the patch that caused the impact.

6. If you double-click on one of the patches in the list, the Patch View for that patch will open.

If no patch impacts have been identified for this product, **File Impacts** will not be listed in the **Impact category** list.
7. To perform patch impact analysis, select the product in the product tree on the **Product View**, and then select Patch Impact Analysis Wizard from the context menu.



Note • All patch information displayed in ConflictSolver comes from the Application Catalog (for imported patches); no information about patches from mssecure.xml file is displayed in ConflictSolver.

Viewing Patch Impacts on the Patches Tab

The information listed on the [Impact Analysis View \(Patches Tab\)](#) depends upon the selection that is made in the **Impact category** list:

- When Summary is selected, details regarding the last time impact analysis was performed are listed.
- When File Impacts is selected, all impacts against this patch are listed, and the product(s) that the patch impacted are identified.



Task:

To view patch impacts:

1. Launch ConflictSolver and click the Patches tab. The Patches Group View opens.
2. Select the patch that you want to examine. The Patch View for that patch opens.
3. Expand the patch to view its constituent views and select the **Impacts** node. The [Impact Analysis View \(Patches Tab\)](#) opens.
4. Select **Summary** from the **Impact category** list to view details regarding the last time impact analysis was performed.
5. Click one of the links to open the [Patch Impact Information Dialog Box](#).

The **Patch Impact Information** dialog box lists the following products on individual tabs:

- Products that were checked for impacts for the selected patch during patch impact analysis.
 - Products that have been updated since the last impact analysis was performed.
 - Products that have been removed from ConflictSolver since the last impact analysis was performed.
 - Products that have been added to the Application Catalog since the last impact analysis was performed.
6. Select **File Impacts** from the **Impact category** list to view a list of all impacts against this patch and the products that was impacted.

If you double-click on one of the listed products, that product will be opened in the **Tables View** with the row that is causing the problem highlighted.

If no impacts have been identified for this patch, **File Impacts** will not be listed in the Impact category list.

7. To perform patch impact analysis, select the patch in the tree, and then select **Patch Impact Analysis Wizard** from the context menu.



Note • All patch information displayed in ConflictSolver comes from the Application Catalog (for imported patches); no information about patches from the mssecure.xml file is displayed in ConflictSolver.

Generating the Patch Report






In Application Manager and ConflictSolver, you can generate a Patch Report which lists detailed information about each patch that has been imported into the Application Catalog, including patch impact analysis information.

The Patch Report is generated in Web Archive format (.mht), a single, stand-alone HTML file that can be easily viewed in a Web browser and copied and emailed throughout your organization. The report is also printer-friendly.



Task: To generate a Patch Report:

1. Launch ConflictSolver or Application Manager and click on the **Patches** tab. The **Patches Group View** opens.
2. Expand the patch listing and select the patch that you want to generate a report on. The **Patch View** opens.
3. Select **Generate Report** from the context menu. The **Save Patch Report As** dialog box opens, prompting you to select a location for the .MHT file.
4. Confirm the report name and location and click **Save**. The report will be generated and will open in a new browser window. Click on the icons to expand or contract that section of the report. The report includes the following sections:

Icon	Name	Description
	General Information	Includes patch title, Microsoft Security Bulletin ID, URL of the Microsoft Security Bulletin, patch summary, and the date the patch was released by Microsoft.
	Products Updated	Products updated by the patch.
	Files	Files included in the patch.
	Registry Entries	Registry entries that are added or modified by the patch.
	Impacts	Products that were checked for impacts during Patch Impact Analysis, and impacts that were detected during Patch Impact Analysis.

5. To print the report with all sections expanded, click the **Print Page** icon.

Reference

This Reference section includes the same topics that are displayed when you click a help button from panel of the Patch Impact Analysis Wizard or OS Security Patch Wizard, or from the Patch Properties dialog box. Reference information is organized into the following areas:

- [Patch Impact Analysis Wizard](#)
- [OS Security Patch Wizard](#)
- [Patch Properties Dialog Box](#)

Patch Impact Analysis Wizard

You can use the Patch Impact Analysis Wizard to identify conflicts between Microsoft operating system patches and the packages and OS Snapshots in your Application Catalog. This helps you determine how specific MSI packages or OS Snapshots would be affected when a Microsoft OS patch is installed.

The Patch Impact Analysis Wizard consists of the following panels:

- [Welcome Panel](#)
- [OS Snapshot Panel](#)
- [Source Patches Panel](#)
- [Target Products Panel](#)
- [Summary Information Panel](#)

When run, the output report is displayed on the **Patch Impact** tab of the ConflictSolver Output Window.

Welcome Panel

The first panel of the Patch Impact Analysis Wizard welcomes you to the Wizard.

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:

Table 15-3 • Patch Impact Analysis Wizard Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific Patch Impact Analysis Wizard panel.

OS Snapshot Panel

On the OS Snapshot Panel, you can optionally select an OS Snapshot to be used to identify specific file information for any patch impacts that are discovered.

The OS Snapshot serves as a representation of the underlying baseline system in your enterprise. If you include an OS Snapshot in your Patch Impact Analysis, then the file version information displayed in warning messages is derived from that OS Snapshot. If an OS Snapshot is not included in the analysis, the file version information will be displayed as “unknown”. As such, *it is recommended to include an OS Snapshot* in your analysis because this allows a finer-tuned evaluation of impacts based upon file and version information.

The following options are included:

Table 15-4 • OS Snapshot Panel Options

Option	Description
Group/OS Snapshot Tree	Select an OS Snapshot to include in your Patch Impact Analysis.

Source Patches Panel

On the Source Patches Panel, you select the patches that you want to include in your analysis.

The following options are included:

Table 15-5 • Source Patches Panel Options

Option	Description
Filter by Product	Lists all the Microsoft operating systems that have a patch that has been imported into the Application Catalog. Make a selection from this list to restrict the list of patches displayed on this panel.
Patch Listing	Patches that meet the selected filter criteria are listed. The following information is provided: <ul style="list-style-type: none">• Id—Number identifying the Microsoft OS patch. Select the check box in this column to include this patch in your analysis.• Name—Name of the patch file.• Title—Description of the purpose of the patch file.• Release Date—Date that this patch was released by Microsoft.
Select All Button	Click to select all listed patches.
Clear All Button	Click to unselect any selected patches.

Table 15-5 • Source Patches Panel Options (cont.)

Option	Description
Patch Properties	Click to access the Patch Properties Dialog Box for this patch, which provides the following information: <ul style="list-style-type: none">• General Tab—Summary information on the patch.• Contents Tab—Listing of the DLL files and registry entries associated with this patch.• Products Tab—A listing of the Product and that product's service packs that this patch is associated with.

Target Products Panel

On the Target Products Panel, select the products or groups of products that you want to perform patch impact analysis on. The Patch Impact Analysis Wizard will analyze the products you select here against the patches you selected on the **Source Patches Panel** for impacts.

The following options are included:

Table 15-6 • Target Products Panel Options

Option	Description
Group/Product Tree	A listing of all groups and products in the open Application Catalog.
Select All Button	Click to select all listed groups and products.
Clear All Button	Click to unselect all selected groups or products.

Summary Information Panel

The Summary Information Panel lists a summary of the options you selected in the Patch Impact Analysis Wizard.

Click **Finish** to begin the Patch Impact Analysis.

OS Security Patch Wizard

The OS Security Patch Wizard consists of the following panels:

- [Welcome Panel](#)
- [Patch File Selection Panel](#)
- [Patch File Branch Selection Panel](#)
- [Patch Bulletin Information Panel](#)

- [Additional Patch Information Panel](#)
- [Patch Import Summary Panel](#)
- [Import Process Results Panel](#)

Welcome Panel

OS Security Patch Wizard is a new AdminStudio tool that is used to import Microsoft operating system patch information into the Application Catalog.

Using the OS Security Patch Wizard, you can open Microsoft patch update files and incorporate the data in those files into the AdminStudio Application Catalog. Click **Next** to continue.

Patch File Selection Panel

On the **Patch File Selection** panel, select a Microsoft operating system patch file that you have downloaded from the [Microsoft TechNet Security Center](#).



Note • The procedure for identifying the Microsoft OS patch files that you want to import, and for downloading them from the Microsoft web site is described in [Identifying and Downloading Microsoft Operating System Patch Files](#). Microsoft operating system patches can be downloaded from the [Microsoft TechNet Security Center](#) at:

<http://www.microsoft.com/technet/security/default.msp>

When you click **Next**, the OS Security Patch Wizard reads and verifies the patch file and progress messages are displayed in the **Processing Status** area of this panel.

- **Prompt to connect to the Application Catalog**—If you opened **OS Security Patch Wizard** from the **Start** menu, after you click **Next**, you will be prompted to provide database connection information. If you opened OS Security Patch Wizard from Application Manager, you will already be connected to the Application Catalog and will not be prompted to connect again.
- **Invalid patch file message**—If you are attempting to import a patch file that is invalid, a message will be displayed and you will be unable to proceed with the import.

When processing of the patch file is successfully finished, one of the following panels opens:

- If you have downloaded a [dual-mode patch](#), the [Patch File Branch Selection Panel](#) opens.
- If the patch you are importing is not a dual-mode patch, the Patch Bulletin Information Panel opens.

Patch File Branch Selection Panel

On the **Patch File Branch Selection** panel, which only opens if you are importing a dual-mode patch, you need to identify the branch of the patch file that you want to import into the Application Catalog.

- **Dual-mode patches**—Windows XP software updates (for example, security updates, critical updates, or hot fixes) that are released after Service Pack 1 (SP1) may include binary files for both Windows XP-based and Windows XP-based SP1 computers. Therefore, there are multiple processing paths for installing this patch. These *dual-mode* update packages are designed to install post-SP1 updates on Windows XP-based computers regardless of whether they are running SP1. If you do not have SP1 installed, the dual-mode update also puts a post-SP1 version of the update on your computer, which is automatically installed if you install SP1. If you remove SP1, the pre-SP1 version of the update is restored.
- **Specifying the branch to import**—On this panel, AdminStudio has preselected the branch of the patch that appears to install the latest version of this patch. You may choose to accept the default selection or make your own selection. It is permissible to import the same dual-mode patch multiple times, selecting different branches each time. Subsequent panels will allow you to uniquely identify each patch branch.

On the **Patch File Branch Selection** panel, the following information is listed for each branch of the patch:

Table 15-7 • Information Displayed for Each Branch of a Dual-Mode Patch


Property	Description
Name	Name identifying each branch of the dual-mode patch.
Branch	<p>The OS Security Patch Wizard imports a Microsoft operating system patch using the patch's INF file, which is a text file that contains all the information necessary to install a device (such as INI and registry changes, filenames, and locations of the source files on source media).</p> <p>The Branch listed in this column is the name of the patch's INF file minus its extension. Because all obvious identifiable information within the INF file (like Name) is often identical, this Branch information is included to help differentiate the patches within the update.</p> <p>A sample branch name could be: update_SP2GDR or update_SP2QFE, where:</p> <ul style="list-style-type: none"> • QFE = Quality Fix Engineering updates (also known as hot fixes). These include security updates, feature packs, update roll-ups, drivers, and critical updates. They are created when an issue is identified that needs a fix in a specific and small subset of customer environments. These fixes have not been tested for the same length of time for regressions or incompatibilities as what a GDR fix typically goes through. • GDR = General Distribution Release updates. These updates can be found on the Windows Update site, meaning that they are available to a wider audience. GDRs are tested to a greater degree than QFEs. <p> Note • QFEs generally contain all the prior fixes from both GDR and QFE releases while GDR releases do not contain QFE fixes.</p>

Table 15-7 • Information Displayed for Each Branch of a Dual-Mode Patch

Property	Description
File Count	Number of files included in the patch.
Registry Count	Number of registry entries included with the patch.

Patch Bulletin Information Panel

From the patch's associated Security Bulletin, copy and paste the **Bulletin ID** and the **Bulletin Title** associated with the patch file you are importing. You should also enter the **Bulletin Summary** and **Bulletin Release Date**.



Important • Security Bulletin data is used by Application Manager to integrate with the Microsoft Web site. Therefore, it is important that you accurately enter this data.



Note • For information on how to view Microsoft Security Bulletins, see [Identifying and Downloading Microsoft Operating System Patch Files](#).

Additional Patch Information Panel

The **Additional Patch Information** panel displays some data that was extracted from the patch file. Review and edit the information and click **Next** to continue.

The following information is included:

Table 15-8 • Information Entered on the Additional Patch Information Panel


Property	Description
Name	By default, this field displays the name of the selected patch file, such as: Windows2000-KB38827-x86-ENU.EXE Edit this name to make it easier for you to identify this patch when it is listed in the Application Catalog.
Associated KB Article	The KB article number is extracted from the patch file and is automatically entered in this field.  Important • The Knowledge Base article identification information is used by Application Manager to integrate with the Microsoft Web site. Therefore, it is important that you accurately enter this data.

Table 15-8 • Information Entered on the Additional Patch Information Panel

Property	Description
Effected Product	Optionally, enter the name of the product that this patch effects. This information is typically something like Windows XP, Windows Vista, etc.
Additional Comments	Comments are extracted from the patch file and automatically entered in this field. You can edit these comments.

Patch Import Summary Panel

The **Patch Import Summary** panel lists the selections you have made in the previous panels of the OS Security Patch Wizard. Click **Import** to initiate the patch import.

Import Process Results Panel

The **Import Process Results** panel lists the results of the import of the selected patch file. The results displayed on this panel, along with the summary information shown in the previous panel, are output to a log file.

To import another patch, click **< Another** to return to the beginning of the OS Security Patch Wizard.

When you have finished importing Microsoft OS patches, click **Finish**.

Patch Properties Dialog Box

You can access the **Patch Properties** dialog box from several locations:

- **Source Patches Panel of the Patch Impact Analysis Wizard**—Select a patch and then click the **Patch Properties** button.
- **Application Manager Patches Tab**—Right-click a patch and then click **Patch Properties**.

The **Patch Properties** dialog box consists of the following tabs:

Table 15-9 • Patch Properties Dialog Box Tabs

Tab	Description
General Tab	View the title and a summary of a selected patch.
Contents Tab	Lists all of the files and registry data contained in the selected patch.
Products Tab	Lists the products that are updated by the selected patch.

General Tab

The **General** tab of the **Patch Properties** dialog box lists the patch **Title** and includes a **Summary** of the purpose of the patch and the patch **Release date**. From the **General** tab, you can also click a link to go directly to the Microsoft Web site and view the Microsoft Security Bulletin and Microsoft Knowledge Base article for that patch:

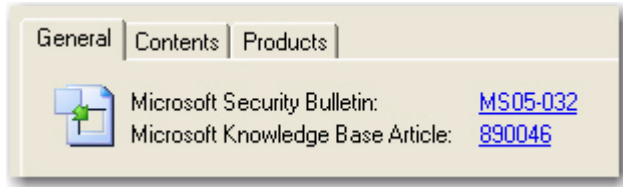


Figure 15-4: Links to Microsoft Security Bulletin and Knowledge Base Article

Contents Tab

The **Contents** tab lists all of the files and registry data contained in the selected patch. The following options are listed:

Table 15-10 • Patch Properties Dialog Box / Contents Tab Options

Option	Description
Files	<p>This section lists all of the files included in the patch. The following information is displayed for each file:</p> <ul style="list-style-type: none"> • File—File name. • Directory—Location where file will be installed. • Version—File version.
Registry data	<p>The registry is a database repository for information about a computer's configuration. This section lists all of the registry data included in the patch. The following information is listed:</p> <ul style="list-style-type: none"> • Key—Name of the registry key. • Name—Name of the registry value. • Value—Data stored for the registry value.

Products Tab

The **Products** tab lists all of the products updated by this patch, and each product’s associated Service Packs. This tab includes the following options:

Table 15-11 • Products Tab Options

Option	Description
Products	All of the products updated by this patch are listed. Select a product from the list to see its associated Service Packs.
Service Packs	Listing of all of the Service Packs associated with the selected product.

Isolating Applications Using Application Isolation Wizard

Application isolation is one solution to component versioning conflicts. Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested.

Application Isolation Wizard user documentation is presented in the following sections:

Table 16-1 • Application Isolation Wizard User Documentation

Section	Description
About Application Isolation Wizard	Explains the reasons you would isolate applications and introduces you to the Application Isolation Wizard.
Launching the Application Isolation Wizard	Explains how to open the Application Isolation Wizard from the AdminStudio interface.
Isolation Methods	Describes the two isolation methods used by Application Isolation Wizard: Manifests and Assemblies, and Windows Installer Isolated Components.
Assemblies	Explains how Assemblies are used.
Manifests	Explains how Manifests are used.
Digital Signatures	Explains how Digital Signatures are used.
Isolating Applications	Describes how to use the Application Isolation Wizard to isolate applications.
Setting Assembly Naming Conventions	Explains assembly naming conventions.

Table 16-1 • Application Isolation Wizard User Documentation (cont.)

Section	Description
Modifying the Default Isolation Recommendations	Describes how to modify the default isolation recommendations when using Windows Installer isolated components and when using manifests for isolation.
Filtering File Listings when Manually Configuring Isolation	Explains how to filter the file listing when manually configuring application isolation using manifests, and when using Windows Installer isolated components.
Servicing Published Shared Assemblies	Explains how to service (alter) shared assemblies after publishing to update them as necessary.
Application Isolation Wizard Reference	Describes Application Isolation Wizard panels and dialog boxes.

About Application Isolation Wizard

Application isolation is one solution to component versioning conflicts, commonly known as “DLL Hell.” Isolation reduces versioning conflicts by modifying an application so it always loads the versions of components—such as DLLs—with which it was originally developed and tested. This is accomplished by providing DLLs and other shared components for specific applications, and placing information traditionally stored in the registry into other files that specify the locations of these isolated components. Application isolation provides increased stability and reliability for applications because they are unaffected by changes caused by installation and ongoing maintenance of other applications on the system.

Depending on the isolation method used in the Application Isolation Wizard, you can partially or totally isolate an application. When using assemblies and manifests to isolate applications for Windows XP systems, the assemblies can be updated following deployment without necessitating application reinstallation.

Reasons to Isolate Applications

You would want to isolate an application if:

- You want to resolve incompatibilities between different versions of shared components.
- You want to reduce the complexity of the installation by storing COM activation data in a manifest instead of the registry.
- You want to insulate the application from changes to shared components.



Tip • Following isolation, you can use the Dynamic Dependency Scanner in InstallShield Editor to verify isolated files are loaded from a different directory.

Reasons Not to Isolate an Application

You would not want to isolate an application if, following application isolation, you discover that the application no longer works because of an internal dependency on a component that has been moved during the isolation process.

Isolating Repackaged Setups Using Repackager

Application Isolation Wizard is a stand-alone tool which accepts a Windows Installer package as input and outputs a new, isolated Windows Installer package. You can also generate an isolated version of a repackaged setup immediately after the build step in Repackager.

If you open a Repackager project and choose the **Create an isolated version of the Windows Installer package** option on the Repackaged Output View, Repackager builds an isolated version of the Windows Installer package immediately after building the non-isolated version.

Both methods of isolating a package are performed using the same Application Isolation Wizard functionality. However, the Application Isolation Wizard provides a user interface experience that allows the user to extend the initial “dependency scanning” process for identifying file isolation candidates, while in Repackager, you specify your assembly and digital signing isolation options on the Isolation Options dialog box, and then those selections are applied to all isolated packages created by Repackager.

For more information, see [Isolating Windows Installer Packages](#).

Launching the Application Isolation Wizard

To launch the Application Isolation Wizard, perform the following steps.



Task: *To launch the Application Isolation Wizard:*

1. Launch AdminStudio.
2. From the Tools Gallery, click the Application Isolation Wizard icon on the left side.



Application Isolation
Wizard

The Application Isolation Wizard launches and you can immediately begin the application isolation process.

Isolation Methods

There are two isolation methods supported by the Application Isolation Wizard™: Manifests and Assemblies and Windows Installer Isolated Components.

Assemblies and Manifests

Application isolation using assemblies and manifests is the recommended isolation method for Windows XP. These assemblies and manifests provide the same end result as Windows Installer isolated components, but keep all information outside of the registry and do not require the components to be installed in the same folder as the application. This reduces the chance of errors after isolation resulting from how the application was written.

Assemblies and manifests only work under the Windows XP operating system.

Windows Installer Isolated Components

Application isolation using Windows Installer isolated components is for Windows 98 SE, Me, and 2000. It can also be used on Windows XP, but using assemblies and manifests is the better solution. The isolated component method copies shared files (typically DLLs) into an application's folder instead of a shared location. The application then uses these files instead of global ones, preventing modifications made by other applications from affecting the shared files. As a result, the application always uses the versions of these files with which it was deployed.

To instruct an application use the private files rather than shared versions, the Application Isolation Wizard populates the IsolatedComponent table with the necessary logic to use private files stored in the same folder as the application. When Windows Installer performs the setup, data from the IsolatedComponent table populates a .local file, which ultimately directs how to use the private files.

Windows Installer isolated components still require some information to be written to the registry, and also require the associated components to be in the same folder as the application. While in most cases this will still provide required isolation, depending on how the application was written, the movement of these associated components from their original locations may prevent the application from functioning correctly.

Assemblies

Assemblies are DLLs or other portable executable files that applications require to function. Under Windows XP, these can be either shared or private. Private assemblies are typically stored in the same directory as the application they support. Shared assemblies are stored in the WinSxS directory, and are digitally signed.

By creating manifests for assemblies, the Application Isolation Wizard™ allows you to create self-contained applications that can use different versions of the same DLL or other portable executable, without any version conflicts.

Shared Assemblies

Shared assemblies are assemblies available to multiple applications on a computer. Applications that require these assemblies specify their dependence within a manifest. Multiple versions of shared assemblies can be used by different applications running simultaneously.

These assemblies are stored in the WinSxS directory, and must be digitally signed for authenticity. After deployment, the version of shared assemblies can be changed, allowing for changes in dependencies.

Private Assemblies

Private assemblies are assemblies created for exclusive use by an application. They are accompanied by an assembly manifest, which contains information normally stored in the registry. Private assemblies allow you to totally isolate an application, eliminating the possibility that dependent files may be overwritten by other applications.

These assemblies are always stored in the same location as their associated executable.

Manifests

The Application Isolation Wizard™ can create two types of manifests: application manifests and assembly manifests.

Table 16-2 • Manifest Types

Manifest Type	Description
Application	<p>Application manifests are XML files that describe an isolated application. This descriptive information includes the relationship between the application and its dependent files.</p> <p>Typically, the naming convention for a manifest is ApplicationName.Extension.manifest. For example, if the application was HelloWorld.exe, the manifest file is called HelloWorld.exe.manifest.</p>
Assembly	<p>Assembly manifests are XML files that describe an application's assemblies. This includes components such as DLLs. Information stored in the assembly manifest, such as COM registration information, ProgIDs, etc., is usually stored in the Registry. However, by making it independent from the registry, only that application can use the dependent files described in the manifest. This enables you to have multiple versions of the same DLL or other portable executable file on a system without generating compatibility conflicts.</p> <p>Typically, the naming convention for a manifest is AssemblyName.Extension.manifest. For example, if the component was Goodbye.dll, the manifest file is called Goodbye.dll.manifest.</p>

Manifests as New Components

When you create manifests, the Application Isolation Wizard supports putting them into new components. If you do not select this option from the Advanced Options dialog box, the manifest will be added to the same component as the assembly.

Digital Signatures

Like conventional signatures, digital signatures identify you (or your organization) to end users. In the context of application isolation, a digital signature identifies you or your organization as the creator of shared assemblies. This ensures that the identity of a shared assembly can be verified for authenticity. Digital signatures in the Application Isolation Wizard™ require a combination of a digital [certificate](#) and a [code signing technology](#).

Certificates

Digital certificates identify you and/or your company to end users to assure them the assembly they are about to use has not been altered. They are issued by a certification authority such as [VeriSign](#), or created using a combination of software publishing credentials (.spc) and a private key (.pvk), both also issued by a certification authority. The certificate includes the public cryptograph key, and, when used in combination with a private key, can be used by end users to verify the authenticity of the signor.

You can create a certificate file from the constituent PVK and SPC files and import it into the [Certificate Store](#) using the [PVK Digital Certificate Files Importer](#). You can then export the certificate (.cer) file for use outside of the certificate store.



Caution • Certificate files must be 2048-bit or higher. For more information, see the article: [Assembly Signing Example](#) on the [Microsoft Developer Network](#) Web site.

Code Signing Technologies

The Application Isolation Wizard™ supports two code signing technologies:

Table 16-3 • Supported Code Signing Technologies

Technology	Description
Credentials	Credentials consist of both Software Publishing Credentials (.spc file) and a private key (.pvk file). These two files are required in conjunction with the certificate to sign shared assemblies.
Certificate Name in the Store	Using Microsoft's Certificate Store technology, the combined software publishing credentials and private key can be placed in a repository for multiple uses. The name of the certificate is provided as opposed to the constituent files in the Credentials code signing technology.

Software Publishing Credentials

You must supply a certification authority with specific information about your company and software to obtain software publishing credentials in the form of an .spc file. Your software publishing credentials are used to generate a digital signature for your assembly.

The .spc file and .pvk ([private key](#)) file you enter in the Digital Signature tab of the Advanced Options dialog box compose the digital certificate for shared assemblies.

Contact a certification authority such as [VeriSign](#) for more information on the specifics of software publishing credentials.

Certificate Store

To perform code signing, both private key and software publishing credential information must be supplied. This must occur each time a package is signed. However, the certificate store serves as a central repository for this information, allowing you to associate the same credentials and key with multiple packages. This simplification is particularly useful when isolating applications, as typically the code signing information will be identical for all shared assemblies. Ultimately, the certificate store removes the burden of managing private key and software publishing credential information.

You can create a certificate file from the constituent PVK and SPC files and import it into the certificate store using the [PVK Digital Certificate Files Importer](#). You can then export the certificate (.cer) file for use outside of the certificate store.

Private Keys

A private key (a file with the extension .pvk) is granted by a certification authority. The Application Isolation Wizard™ uses the private key you enter in the Digital Certificates tab of the Advanced Options dialog box to digitally sign your shared assembly and ensure end users of its content's authenticity.

The .spc ([Software Publishing Credentials](#)) file and .pvk file you enter in the Digital Signature tab compose the digital certificate for shared assemblies.

Contact a certification authority such as [VeriSign](#) for more information on the specifics of software publishing credentials.

Isolating Applications

To isolate applications within a Windows Installer package or a merge module, perform the following steps.



Task: *To isolate applications within a Windows Installer package (.msi) or merge module (.msm):*

1. Launch the Application Isolation Wizard™. The **Welcome Panel** appears.
2. From the **Welcome Panel**, click **Next**. The **Windows Installer File Selection Panel** appears.
3. From the Windows Installer File Selection Panel, specify the Windows Installer package (.msi), Windows Installer self-extracting executable file (setup.exe), merge module (.msm) containing applications you want to isolate. Click **Next**. The Isolation Method panel appears.
4. From the **Isolation Method Panel**, select the isolation method(s) you want to use.
5. If you are using manifests, you can click Advanced to configure manifest properties and digital signature information (if required) on the [Advanced Options Dialog Box](#).
6. Click **Next**. The **Summary Information Panel** appears.
7. From the **Summary Information Panel**, confirm the isolation selections.
8. If you want to manually configure isolation, click Modify.
 - **If you are using manifests** to isolate your application—either alone or in conjunction with Windows Installer isolated components—the **Manifest and Assembly Design** dialog box appears.
 - **If you are only using Windows Installer isolated components** to isolate the application, the **Isolated Components Design** dialog box appears. After you have completed manually configuring the isolation, click OK to return to the Summary Information Panel.
9. Click Isolate. The **Application Isolation Progress Panel** appears.

When the Application Isolation Wizard is complete, the **Completing the Application Isolation Wizard Panel** is displayed, providing feedback on whether the Application Isolation Wizard was successful.
10. From the **Completing the Application Isolation Wizard Panel**, click **Finish**.

Setting Assembly Naming Conventions

To set the default naming convention for assemblies, perform the following steps.



Task: *To set the default naming convention for assemblies:*

1. Launch the Application Isolation Wizard™. The **Welcome Panel** opens.
2. From the Welcome panel, click Next. The **Windows Installer File Selection Panel** opens.
3. From the Windows Installer File Selection panel, specify the Windows Installer package (.msi) or merge module (.msm) containing applications you want to isolate. Click Next. The **Isolation Method Panel** opens.
4. Select the Use manifests for isolation option.
5. Click Advanced. The Manifest Options tab of the **Advanced Options** dialog box opens.

6. Enter your Company name and Division. These two fields create the default assembly naming convention (in the form “Company.Division.Assembly” followed by a number).



Note • To edit the Assembly Name, you can also click **Modify** from the **Summary Information Panel** later in the Wizard to open the **Manifest and Assembly Design** dialog box, and then click **Properties** to open the **Application Manifest Properties** dialog box, where you can edit the Assembly Name.

7. Click **OK**. You are returned to the **Isolation Method Panel**.
8. Click **Next**. The **Summary Information Panel** opens.
9. Click **Isolate** to proceed with isolation using the specified naming convention.

Assemblies created during application isolation will follow the naming convention as specified.

Modifying the Default Isolation Recommendations

You can modify default isolation recommendations for the following:

- [When Only Using Windows Installer Isolated Components](#)
- [When Using Manifests for Isolation](#)

When Only Using Windows Installer Isolated Components

To modify the default isolation recommendations when only using Windows Installer isolated components, perform the following steps.



Task: *To modify the default isolation recommendations:*

1. From the **Summary Information Panel** of the Application Isolation Wizard™, click **Modify**. The **Isolated Components Design** dialog box is displayed.
2. Select the Applications to be Isolated and then select the Files to Isolate for Selected Application. Repeat as necessary.
3. Click **OK**. When you return to the Summary Information Panel, verify your settings before isolating.

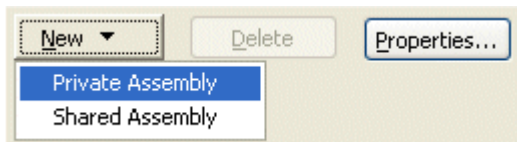
When Using Manifests for Isolation

To modify the default isolation recommendations when using manifests for isolation, perform the following steps.



Task: *To modify the default isolation recommendations when using manifests for isolation:*

1. From the **Summary Information Panel** of the Application Isolation Wizard, click Modify. The **Manifest and Assembly Design** dialog box is displayed.
2. If you need to create a new assembly, select the application for which you want to create the assembly, click New, and select the assembly type: Private Assembly or Shared Assembly.



A new assembly is listed under the selected application.

3. Select the new assembly then add or remove files in the assembly.
4. Click OK. When you return to the Summary Information panel, verify your settings before isolating.

Filtering File Listings when Manually Configuring Isolation



Task: *To filter the file listing when manually configuring application isolation using manifests:*

1. In the **Manifest and Assembly Design** dialog box, select an assembly from the application tree.
2. Directly below the Files to be added in an assembly window, specify the files displayed from the Show filter.



Task: *To filter the file listing when manually configuring application isolation using Windows Installer isolated components:*

1. In the **Isolated Components** dialog box, select an application from the application tree.
2. Directly below the Files to Isolate for Selected Application window, specify the files displayed from the Show filter.

Servicing Published Shared Assemblies

Shared assemblies can be serviced (altered) after publishing to update them as necessary. This is accomplished using a publisher configuration, which overrides default configurations specified in the manifest.

For an exhaustive discussion of how to service shared assemblies, refer to the article [How To Build and Service Isolated Applications and Side-by-Side Assemblies for Windows XP](#) on the Microsoft Developer Network Web site (msdn.microsoft.com).

Application Isolation Wizard Reference

The Application Isolation Wizard scans Windows Installer packages (.msi) or merge modules (.msm) and isolates applications within them. Isolation ensures that applications always use the specific shared and support files with which they were installed. This prevents the overwriting of previous versions of shared components, and ensures that other applications do not overwrite the versions of shared and support files required by your application.

The following reference topics are available for the Application Isolation Wizard:

- [Welcome Panel](#)
- [Windows Installer File Selection Panel](#)
- [Isolation Method Panel](#)
- [Summary Information Panel](#)
- [Application Isolation Progress Panel](#)
- [Completing the Application Isolation Wizard Panel](#)
- [Advanced Options Dialog Box](#)
- [Manifest and Assembly Design Dialog Box](#)
- [Isolated Components Design Dialog Box](#)
- [Assembly Properties Dialog Box](#)
- [Application Manifest Properties Dialog Box](#)

Welcome Panel

The Welcome panel is the first panel displayed when you launch the Application Isolation Wizard. It provides a general explanation of application isolation.

Click Next to proceed to the **Windows Installer File Selection Panel**.

Windows Installer File Selection Panel

Enter the full path and file name of the Windows Installer package (.msi) or merge module (.msm) that you want the Application Isolation Wizard to scan for isolation candidates. Alternately, click Browse to navigate to the file.


Click Back to return to the **Welcome Panel**; click Next to proceed to the **Isolation Method Panel**.

Isolation Method Panel

Select the application isolation method(s) you want to use for this Windows Installer package or merge module.

Make the following selections under **Manifests** and **Windows Installer Isolated Components**:

Table 16-4 • Isolation Method Panel Selection Guidelines

If Installation Will Be Deployed ...	Manifests	Windows Installer Isolated Components
Only on Windows 2003 Server	Select	Do Not Select
Only on Windows XP	Select	Do Not Select
Windows 98 SE, Me, and/or Windows 2000 but not XP	Do Not Select	Select
Windows XP and Windows 98 SE, Me, and/or Windows 2000  Note • Your installation package size will increase, but application isolation will work on the appropriate operating systems.	Select	Select

If you use **Manifests**, click **Advanced** to display the **Advanced Options** dialog box, from which you can configure manifest options and digital signature information.

Summary Information Panel

From this panel, review a summary of your selections prior to isolation.

For granular control over the isolation process, click Modify. If you are only using Windows Installer Isolated Components as the isolation method, the **Isolated Components Design** dialog box appears. Otherwise, the **Manifest and Assembly Design** dialog box is displayed.

Click Back to return to the **Isolation Method Panel**; click Isolate to isolate the application according to your settings. The Application Isolation Progress panel is displayed.

Application Isolation Progress Panel

During application isolation, the progress is displayed on this panel. Information about the applications, assemblies (if using manifests as the isolation method), and files is displayed above the progress bar.

Upon isolation completion (or failure), the **Completing the Application Isolation Wizard Panel** is displayed.

Completing the Application Isolation Wizard Panel

The final panel in the Application Isolation Wizard provides feedback on whether the Application Isolation Wizard was successful.

If the Application Isolation Wizard was successful, the names and locations of the original and output packages are provided. If the Wizard was not successful, this panel informs you that the selected components could not be isolated.

Advanced Options Dialog Box

The **Advanced Options** dialog box, available from the **Isolation Method** and **Ready to Isolate** panels of the Application Isolation Wizard, allows you to configure assembly types, naming conventions, and digital signature options. The **Advanced Options** dialog box presents these options on two tabs: **Manifest Options** and **Digital Signature**.




Once you have finished configuring advanced options, click OK to save your changes, or Cancel to close the dialog box without saving your modifications. When the dialog box closes, you are returned to the panel where you clicked Advanced.

Manifest Options Tab

The Manifest Options tab, available in the **Advanced Options** dialog box, allows you to configure several settings associated with manifests.

These settings include:

Table 16-5 • Manifest Options Tab Options

Option	Description
Assembly Type	<p>This option allows you to select the type of assemblies that Application Isolation Wizard™ will create and use:</p> <ul style="list-style-type: none"> • Create private side-by-side assemblies in the application folder • Create shared side-by-side assemblies in the WinSxS folder (Default) <p>If you want to use both assembly types, you need to manually configure assemblies from the Manifest and Assembly Design Dialog Box.</p>  <p>Note • Manifests for shared assemblies must be digitally signed. This can be done in the Digital Signature Tab.</p>  <p>Note • A 2048-bit key is required to sign a Windows XP assembly/manifest being installed to the WinSxS folder.</p>
Assembly Naming Conventions	<p>Specify your company and division information to define the default naming convention that Application Isolation Wizard will use when creating assemblies during application isolation</p> <p>By default, assembly names are specified in the form of:</p> <p>Company.Division.Assembly</p>
Create a new component for each assembly	<p>Select this option if you want to create a new component for each assembly created during isolation.</p> <p>This check box applies to all assemblies created. Individual assemblies can be configured from the Assembly Properties dialog box on a per-assembly basis.</p>  <p>Caution • If you are creating assemblies for applications files within multiple components, this option must be selected for successful application isolation.</p> <p>If you are planning to deploy this isolated package to operating systems prior to Windows XP, always check this box.</p>

Digital Signature Tab



The Digital Signature tab, available in the **Advanced Options** dialog box, allows you to configure the certificate information required when using shared assemblies. This required digital signature provides an extra layer of protection, allowing you to obtain information about the company who created a global assembly.



Caution • The Application Isolation Wizard™ uses timestamping when signing global assemblies. Consequently, you must have an Internet connection on the computer when you create a global assembly.

You must configure the following options when signing these assemblies:

Table 16-6 • Digital Signature Tab Options

Option	Description
Certificate File	<p>Click the Browse () button next to the field and navigate to the certificate file you are using to sign assemblies.</p> <p>A digital certificate identifies you and/or your company to end users and assures them the data they are about to receive has not been altered.</p>
Credentials	<p>Select this option to use credential files as the code signing technology. If you select this option, you must supply the name and location of both your software publishing credential files: SPC File and PVK File.</p> <p></p> <p>Note • In order to receive a software publishing credentials and a private key, you must supply a certification authority, such as such as VeriSign, with specific information about your company and software.</p>
SPC File	Specify the name and location of your software publishing credentials file (.spc).
PVK	Specify the name and location of your private key file (.pvk).
Certificate Name in the Store	Select this option to use an existing certificate file in the Certificate Store as the code signing technology. The Certificate Store is a central repository for certificate files. Using a Certificate Store allows you to reuse the certificate files for different purposes as necessary.



Note • A 2048-bit key is required to sign a Windows XP assembly/manifest being installed to the WinSxS folder.

Manifest and Assembly Design Dialog Box

If you are using manifests to isolate your application, either alone or in conjunction with Windows Installer isolated components, the Manifest and Assembly design dialog box is displayed when you click Modify from the Summary Information panel.

When you first display this dialog box, the settings the Application Isolation Wizard™ recommends for this package are displayed. By default, only executables that will be installed in the SystemFolder will be selected for isolation. You can select an application contained in the Windows Installer or merge module and create a new private or shared assembly for that application. You can then select the files to isolate for the selected application. A filter at the bottom of the dialog box allows you to restrict the file types visible.

Click Properties to display the **Application Manifest Properties** dialog box. From this dialog box, you can configure the naming convention for assemblies and manifests, and specify whether you want manifests placed into separate components.

When you have finished performing manual configuration, click OK to return to the Summary Information panel.

Isolated Components Design Dialog Box

If you are only using Windows Installer isolated components to isolate the application, this dialog box is displayed when you click Modify from the Summary Information panel.

When you first display this dialog box, the settings the Application Isolation Wizard™ recommends for this package are displayed. By default, only libraries that will be installed in the SystemFolder will be selected for isolation. You can select an application contained in the Windows Installer or merge module, and then select the files to isolate for the selected application. A filter at the bottom of the dialog box allows you to restrict the file types visible.

When you have finished performing manual configuration, click OK to return to the Summary Information panel.

Assembly Properties Dialog Box


The Assembly Properties dialog box displays information about the manifest and assembly, and can be launched from the Manifest and Assembly Design dialog by selecting an assembly and clicking Properties.

The following groups contain configurable options:

Table 16-7 • Assembly Properties Dialog Box Options

Group	Description
Manifest Details	In the Manifest Details group, you can view the file name for the manifest. It is structured in the form "Company.Division.Assembly.manifest" by default.
Assembly Identity	The Assembly Identity group contains fields for the Assembly Name and Version. When you change the assembly name, the manifest file name changes.

Table 16-7 • Assembly Properties Dialog Box Options (cont.)

Group	Description
Assembly Type	This group allows you to select whether the current assembly is private or shared. If it is shared, you must configure digital signature information in the Advanced Options dialog box.
Create new component	<p>Select this option if you want to create a new component for this assembly.</p>  <p>Caution • If this assembly contains files that originate from multiple components, this option must be selected for successful application isolation.</p> <p>If you are planning to deploy this isolated package to operating systems prior to Windows XP, always check this box.</p> <p>This check box applies only to this assembly. Global settings for assemblies can be configured from the Manifest Options tab of the Advanced Options dialog box.</p>

Application Manifest Properties Dialog Box

The Application Manifest Properties dialog box displays information about the manifest and assembly, and can be launched from the Manifest and Assembly Design dialog by selecting an application and clicking Properties.

The following groups contain configurable options:

Table 16-8 • Application Manifest Properties Dialog Box Options

Options Group	Description
Manifest Details	In the Manifest Details group, you can view the file name for the manifest. It is structured in the form “Company.Division.Assembly.manifest” by default.
Assembly Identity	The Assembly Identity group contains fields for the Assembly Name and Version. When you change the assembly name, the manifest file name changes.

Command-Line Options

The Application Isolation Wizard can also be run from the command line. You can specify the following options when running the AIW.exe executable from the command line:

Table 16-9 • Application Isolation Wizard Command-Line Options

Option	Description
-?	Displays command-line help for the Application Isolation Wizard.
-version	Displays the version of AdminStudio.
-i <configuration file>	Allows you to specify a configuration file for Application Isolation Wizard settings. The default file, AIWConfig.ini, is located in <AdminStudio Directory>\Common and can be used as a model. This parameter is optional.
-p <package name>	The name and location of the package or merge module which includes applications you want to isolate. This parameter is mandatory.

Configuration Files

When using the command-line options for the Application Isolation Wizard, you can specify an INI file for configuration using the **-i** parameter. This file should take the following format:

```
[IsolationMethods]
Manifests=1
IsolatedComponents=1

[DigitalSignature]
CertificateFile=
SPCFile=
PVKFile=
CertificateName=
TimeStampAssemblies=

[Manifest]
AssemblyType=0
Company="Company"
Division="Division"
NewComponents=0
```

Each configuration corresponds to a user interface setting in the Application Isolation Wizard, as described below:

Table 16-10 • Configuration File Settings

INI File	UI Setting	Explanation
Manifests	Use manifests for isolation option on Isolation Method panel	Set this value to 1 to use manifests. Manifests only work with Windows XP.

Table 16-10 • Configuration File Settings (cont.)

INI File	UI Setting	Explanation
IsolatedComponents	Use Windows Installer isolated components for isolation on Isolation Method panel	Set this value to 1 to use Windows Installer isolated components.
CertificateFile	Certificate File field on the Digital Signature tab of the Advanced Options dialog box	Provide the name and location of the CER file.
SPCFile	SPC File field on the Digital Signature tab of the Advanced Options dialog box	Provide the name and location of the SPC file
PVKFile	PVK File field on the Digital Signature tab of the Advanced Options dialog box	Provide the name and location of the private key.
CertificateName	Certificate Name in the store field on the Digital Signature tab of the Advanced Options dialog box	Provide the name of the certificate from the certificate store.
TimeStampAssemblies	No corresponding UI setting	Set this value to 0 to disable timestamping during shared assembly creation; set it to 1 to enable timestamping. By default, the Application Isolation Wizard uses timestamping if this value is not configured.
AssemblyType	Assembly Type on the Manifest Options tab of the Advanced Options dialog box	Set this value to 0 to use private assemblies; set the value to 1 to use shared assemblies.
Company	Company field on the Manifest Options tab of the Advanced Options dialog box	Put the name of your company in quotes.
Division	Division field on the Manifest Options tab of the Advanced Options dialog box	Put the name of your division in quotes.
NewComponents	Create new component option in Manifest Options tab of Advanced Options dialog box	Set the value to 1 to create a new component for each manifest.

Manifest Examples

Following are examples of both an application manifest and an assembly manifest:

Application Manifest Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <assembly xmlns="urn:schemas-microsoft-com:asm.v1" manifestVersion="1.0">
<assemblyIdentity type="win32" name="InstallShield.Development. AppAssembly4" version="1.0.0.1"
processorArchitecture="x86" />
<description>This manifest was generated by the Application Isolation Wizard</description>
- <dependency>
  - <dependentAssembly>
    <assemblyIdentity type="win32" name="InstallShield.
      Development.LocalAssembly1" version="1.0.0.1"
      processorArchitecture="x86" />
  </dependentAssembly>
</dependency>
</assembly>
```

Assembly Manifest Example

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
- <assembly xmlns="urn:schemas-microsoft-com:asm.v1"
  manifestVersion="1.0">
  <assemblyIdentity type="win32" name="InstallShield.Development.
    LocalAssembly1" version="1.0.0.1" processorArchitecture="x86" />
  <file name="IsCommonServices.dll">
- <comClass description="CabinetBuilder Class" clsid="
  {8D3FE200-DA96-11D3-BEE7-00105A996B4E}" progid=
    "ISHerculesCommonServices.CabinetBuilder.1"
    threadingModel="Apartment" tlbid="{2491C036-D5B0-11D3-BEE5-
      00105A996B4E}">
    <progid>ISHerculesCommonServices.CabinetBuilder</progid>
  </comClass>
- <comClass description="Cabinet Class" clsid="
  {3C35E807-C92D-11D3-BEDF-00105A996B4E}" progid=
    "ISHerculesCommonServices.CabinetExtractor.1"
    threadingModel="Apartment" tlbid="{2491C036-D5B0-11D3-BEE5-
      00105A996B4E}">
    <progid>ISHerculesCommonServices.CabinetExtractor</progid>
  </comClass>
- <comClass description="InstallShield Common Services Registry
  object" clsid="{3032B526-2C3D-11D4-AB2C-00C04F09719A}"
  progid="ISHerculesCommonServices.Registry.1" threadingModel=
    "Apartment" tlbid="{2491C036-D5B0-11D3-BEE5-00105A996B4E}">
    <progid>ISHerculesCommonServices.Registry</progid>
  </comClass>
</file>
</assembly>
```


Ensuring Package Quality Using QualityMonitor



Edition • *QualityMonitor is included with AdminStudio Professional and Enterprise Editions.*

QualityMonitor allows you to run a series of built-in tests to installed Windows Installer-based products, helping to ensure they run correctly. When failures occur, QualityMonitor can help identify where problems exist, and ultimately direct you to the solution.

QualityMonitor user documentation is presented in the following sections:

Table 17-1 • QualityMonitor User Documentation

Section	Description
About QualityMonitor	Explains the purpose and benefits of using QualityMonitor.
Creating New QualityMonitor Project Files	Explains how to create a new QualityMonitor project.
Opening Existing QualityMonitor Project Files	Explains how to open QualityMonitor project files, which have an .iqm extension.
Working with Test Cases	Describes the most common tasks you may perform when working with Test Cases.
Deployment Testing	Explains how to perform deployment tests against the installed product, ensuring that the product has been installed correctly.
Lockdown and Runtime Testing	Explains how to test an application when its target environment is restricted in some way, such as in a locked-down environment.

Table 17-1 • QualityMonitor User Documentation (cont.)

Section	Description
Using MSI Doctor to Verify Package Deployment Status	Explains how to use MSI Doctor to verify if an MSI package is installed properly. This helps prevent users from seeing an auto-repair dialog box when they run the application.
Creating Custom Test Cases	Explains how to add additional, custom Test Cases to projects—based on your business needs.
Test Reports	Explains how to create an HTML test report for the current project.
Running QualityMonitor from the Command Line	Explains how to run QualityMonitor from the command line.
QualityMonitor Reference	Provides detailed reference on each user interface element, dialog box, and view in QualityMonitor.

About QualityMonitor

Prior to deploying a Windows Installer–based application, typically you need to test it in the targeted deployment environment to ensure the application works as expected. However, it is often not feasible (or possible) to test each piece of an application’s functionality, due to the complexity of the application and/or its interface. Behind the scenes, there may be dozens or hundreds of attempts to access files, registry keys, or services; errors may only become apparent in rare and isolated circumstances.

One major source of failure is when the target environment is restricted in some way, such as in a locked-down environment. In this case, there may be prohibitions on certain COM activation or registry access, which ultimately prevents an application from working correctly.

QualityMonitor allows you to run a series of built-in tests to installed Windows Installer-based products, helping to ensure they run correctly. When failures occur, QualityMonitor can help identify where problems exist, and ultimately direct you to the solution.

Creating New QualityMonitor Project Files

You can create a new QualityMonitor project by selecting the **Create new project** option on the **Welcome to QualityMonitor View**.



Task: *To create a new QualityMonitor project file (.iqm):*

1. Launch QualityMonitor. The **Welcome to QualityMonitor View** opens.
2. Click **Open** on the **File** menu. The **Open QualityMonitor Project** dialog box opens.

3. Select the **Select an application that is installed on this machine ...** option.
4. Select an application from the available applications list.
5. Click **OK**. A new QualityMonitor project for the selected application is opened, and the **Product Information View** opens.

Opening Existing QualityMonitor Project Files

QualityMonitor's project files have an `.iqm` extension and are opened by performing the following steps.



Task: *To open an existing QualityMonitor project file (.iqm):*

1. Launch QualityMonitor. The **Welcome to QualityMonitor View** opens.
2. Click **Open** on the **File** menu. The **Open QualityMonitor Project** dialog box opens.
3. Select the **Open QualityMonitor project (.iqm) file** option.
4. Enter or browse to the file you want to open.
5. Click **OK**.

Working with Test Cases

The primary purpose of QualityMonitor is to serve as a diagnostic tool when applications fail to function correctly in deployment environments. This is accomplished primarily through running Test Cases and individual Test Items, and evaluating the results. Topics in this section cover the most common tasks you may perform when working with Test Cases.

- [Running Individual Test Items](#)
- [Running Multiple Test Items](#)
- [Adding Test Item Comments](#)
- [Adding Test Case Comments](#)
- [Viewing Test Item Details](#)
- [Clearing Test Case Results](#)
- [Manually Setting Test Case Status](#)
- [Manually Setting Test Item Status](#)
- [Filtering Test Case Data](#)

Running Individual Test Items

You can choose to run a **Deployment Test** on an individual **Test Item**.



Task: *To run an individual Test Item:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. In the **Test Items** list, select the **Test Item** you want to run and click the Run button.

Depending on whether the **Test Item** is automatic or requires opening or launching files, you may need to perform some manual tasks prior to results being returned. When the test is complete, QualityMonitor displays **Passed** or **Failed** in the **Status** column of the selected **Test Item**.

Running Multiple Test Items

You can choose to run a **Deployment Test** on a multiple **Test Items** at once.



Task:

To run multiple Test Items:

1. Create or open a QualityMonitor project. The **QualityMonitor Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. In the **Test Items** list, select the **Test Items** you want to run. Multiple selection is supported using the Shift and Ctrl keys.
4. To run only the selected **Test Items**, click the **Run** button. To run all **Test Items**, click the **Run All** button.

Depending on whether the **Test Item** is automatic or requires opening or launching files, you may need to perform some manual tasks prior to results being returned. When the tests are complete, QualityMonitor displays **Passed** or **Failed** in the **Status** column of the selected **Test Items**.

Adding Test Item Comments

You can choose to add comments to a **Test Item**—perhaps to document why it passed or failed, or to note an issue that needs attention.



Task: *To add comments to an individual Test Item:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. In the **Test Items** list, right-click the **Test Item** to which you want to add comments, and select **Test Item Information** from the context menu. The **Test Item Information** dialog box appears.
4. Enter comments into the **Comments** field.
5. When finished, click **OK**.

Adding Test Case Comments

You can add comments associated with an entire Test Case.



Task: *To add comments to a Test Case:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. In the **Comments** box at the top right of the View, enter comments. Your comments are automatically saved.

Viewing Test Item Details

When a Test Item fails, you can view details about it, including the error message associated with it. This information is displayed on the **Test Item Information** dialog box, along with the Test Item name, status, and any comments that have been entered.



Task: *To view Test Case details:*

1. Create or open a QualityMonitor project. The **QualityMonitor Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**

- **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. Right-click on a **Test Item** and select **Test Item Information** from the context menu. The **Test Item Information** dialog box opens, and the following details are listed:
- **Test Item**—Name of the selected Test Item.
 - **Status**—Status of the selected Test Item: Passed, Failed, or Pending.
 - **Comments**—Any comments that were previously entered.
 - **Test Details**—If this Test Item has **Failed**, a brief explanation of the reason the Test Item failed the test is listed.
4. When finished viewing test details, click **OK** to close the dialog box.

Clearing Test Case Results

You can clear Test Case results for all Test Items from a previous Test Case execution.



Task: *To clear Test Case results (all Test Items) from a previous Test Case execution:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**

3. Click the **Reset Results** button.

When you click **Reset Results**, the status of *all* Test Items is reset. To reset the status of one individual **Test Item**, right-click that **Test Item**, point to **Set Status** and select **Pending** from the context menu. See [Manually Setting Test Case Status](#).

Manually Setting Test Case Status

Depending on your business practices and standards, you may want to override the status of a Test Case in the View List from its current state. In most cases, this will be setting a Test Case which QualityMonitor has marked as **Failed** (because one or more individual Test Items have failed) to **Passed**.



Task:***To manually set the Test Case status:***

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. In the **Test Case Status** area, change the status to the desired state by selecting **Pending**, **Passed**, or **Failed**.

Manually Setting Test Item Status

You can manually set the status of an individual Test Item to Passed, Failed, or Pending.



Task: *To manually set the status of an individual Test Item:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**
 - **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. Right-click on the **Test Item**, point to **Set Status**, and select the status from the Set Status submenu: **All**, **Pending**, **Passed**, or **Failed**.

Filtering Test Case Data

On the **Product Information View**, you can choose to display only those Test Cases with a selected status: **Pending**, **Passed**, or **Failed**.



Task: *To filter the displayed data in a Test Case:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens and a view of **Test Items** is displayed.
2. Under **Deployment Tests**, select one of the following tests to open its corresponding View:
 - **Class IDs**
 - **File Associations**
 - **Help Files**
 - **Prog IDs**

- **Services**
 - **Shortcuts**
 - **Type Libraries**
 - **ODBC Data Sources**
 - **ODBC Drivers**
3. From the **View these test items** list, select the filter you want to apply to the data: **All**, **Passed**, **Failed**, or **Pending**. The View is automatically updated based on the selected filter.

Deployment Testing

Deployment tests are performed against the installed product, ensuring that the product has been installed correctly and all key functionality works in the installed environment. Test Cases in this area are primarily designed to identify whether the application fails to work properly due to permission settings on the registry or individual files.

Some of the primary areas checked are:

Table 17-2 • Areas Checked During Deployment Testing

Area	Description
COM Data	Ensure all COM objects can be instantiated programatically. This includes Class IDs, Prog IDs, and Type Libraries. COM data is tested silently, returning results in the Test Case Progress area and the queue. See Checking Class IDs , Checking Prog IDs , or Checking Type Libraries .
File Associations	Ensure all file extensions have been installed and associated correctly. This involves launching a file with this extension, and determining if the correct application was used. See Checking File Associations .
Help Files	Ensure help files are installed and can be launched correctly. See Checking Help Files .
Shortcuts	Ensure each shortcut is installed and if it successfully launches the shortcut target. See Checking Shortcuts .
Type Libraries	Determines if the Type Libraries COM objects can be instantiated programatically. See Checking Type Libraries .
Manifests	Tests the manifests and assemblies used to isolate a Windows Installer package. See Checking Manifests .
ODBC Data Sources	Verify ODBC data sources. See Checking ODBC Data Sources .
ODBC Drivers	Verify ODBC drivers. See Checking ODBC Drivers .

Table 17-2 • Areas Checked During Deployment Testing (cont.)

Area	Description
Services	Ensure all NT Services have been installed correctly. This is done by opening the Services Manager to determine if the Service exists on the target machine. See Checking Services .

Automatically Running All Deployment Tests Silently

You can choose to run all deployment tests silently (without prompting for user input) using either the Interface or the command line.

From the Interface

You can choose to run all deployment tests silently (without prompting for user input) by making a selection in the QualityMonitor interface.



Task: *To run all deployment tests silently from the Interface, do one of the following:*

1. On the QualityMonitor **Product Information View**, select the **Deployment Tests** root node and then do one of the following:
 - Click the **Execute All Deployment Tests** button.
 - Select **All Deployment Tests** from the **Execute** menu.
 - Click the **Execute All Deployment Tests** toolbar button:



When you select one of these options, a dialog box with a progress bar and an option to cancel will be displayed.

From the Command Line

You can also run all deployment tests silently by entering a command in the command line. See [Running QualityMonitor from the Command Line](#) for more information.

Checking Class IDs

The **Class ID** Deployment Test is run to determine if the Class ID COM objects can be instantiated programmatically. COM data is tested silently, returning results in the **Test Case Progress** area and the queue.



Task: *To check Class ID functionality:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Class IDs**. The **Class IDs View** opens.

3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When testing is finished, results are listed in the **Test Case Progress** area.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

Checking File Associations

The **File Associations** Deployment Test is run to determine if all file extensions have been installed and associated correctly. This involves launching a file with this extension, and determining if the correct application was used.



Task: *To test file associations:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **File Associations** from the View List. The **File Associations View** opens.
3. Right-click on the Test Item you want to run and select **Run**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When the **Test Progress** dialog box opens, click **Run** to exercise the file association.
5. From the resulting **Open** dialog box, browse to a file with the appropriate extension and click **Open**. The file is launched with its associated application. Following the application launch, the **Test Result** dialog box appears.
6. Click **Yes** or **No** depending on whether the file launched with the expected program. You can also enter comments in the **Comment** field on this dialog box.

Checking Help Files

The **Help Files** Deployment Test is run to determine if the help files are installed and can be launched correctly.



Task: *To test help file functionality:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Help Files** from the View List. The **Help Files View** opens.

3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When the **Test Progress** dialog box opens, click **Run** to launch the help file. Following an attempt to launch the shortcut, the **Test Result** dialog box appears.
5. Click **Yes** or **No** depending on whether the help file launched correctly. You can also enter comments in the **Comment** field on this dialog box.

Checking Prog IDs

The **Prog IDs** Deployment Test is run to determine if the Prog ID COM objects can be instantiated programatically. COM data is tested silently, returning results in the **Test Case Progress** area and the queue.



Task: To check Prog ID functionality:

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Prog IDs** from the View List. The **Prog IDs View** opens.
3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When testing is finished, results are recorded in the **Test Case Progress** area.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

Checking Services

The **Services** Deployment Test is run to determine if all NT Services have been installed correctly. This is done by opening the Services Manager to determine if the Service exists on the target machine.



Task: To check Service functionality:

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Services** from the View List. The **Services View** opens.
3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When testing is finished, results are recorded in the **Test Case Progress** area.



Note • When a **Test Item** is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the **Test Item** and then select **Test Item Information** from the context menu.

Checking Shortcuts

The **Shortcuts** Deployment Test is run to determine if each shortcut is installed and if it successfully launches the shortcut target.



Task: *To check shortcuts:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Shortcuts** from the View List. The **Shortcuts View** opens.
3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When the **Test Progress** dialog box opens, click **Run** to launch the shortcut. The **Test Result** dialog box opens.
5. Following an attempt to launch the shortcut, the **Test Result** dialog box opens. Click **Yes** or **No** depending on whether the shortcut launched correctly. You can also enter comments in the **Comment** field on this dialog box.

Checking Type Libraries

The **Type Libraries** Deployment Test is run to determine if the Type Libraries COM objects can be instantiated programatically. COM data is tested silently, returning results in the **Test Case Progress** area and the queue.



Task: *To check type library functionality:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Type Libraries** from the View List. The **Type Libraries View** opens.
3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.
4. When testing is finished, results are recorded in the **Test Case Progress** area.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

Checking Manifests

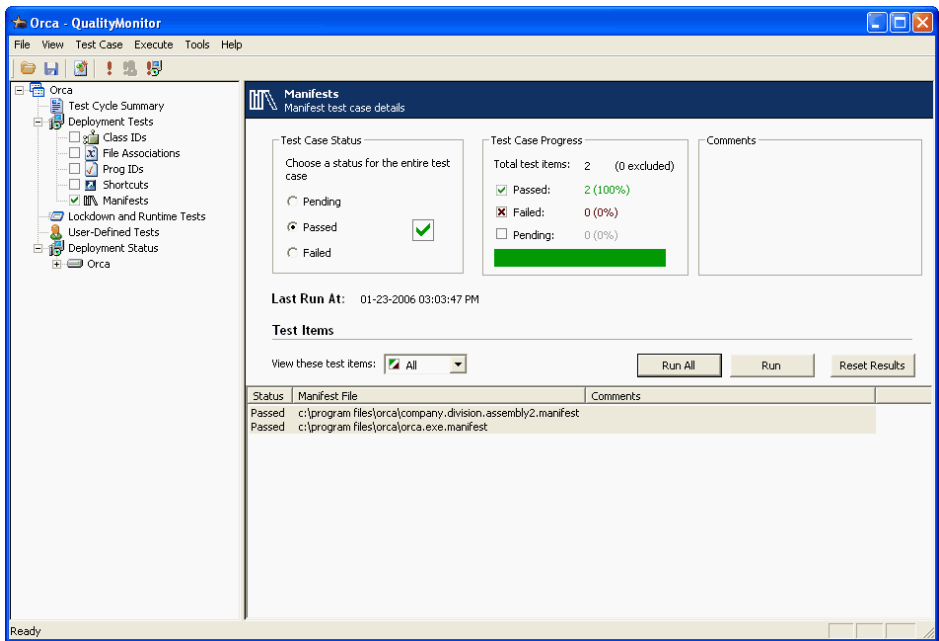
The **Manifests** Deployment Test is run to test the manifests and assemblies used to isolate a Windows Installer package.

The Manifests Deployment Test tests information from the MsiAssembly and MsiAssemblyName tables. QualityMonitor reads through the manifest/assembly files and performs the baseline Class IDs, Prog IDs, or Type Libraries testing for each entry in the files.



Task: *To check shortcuts:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **Manifests** from the View List. The **Manifests View** opens.



3. Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.

When testing is finished, results are recorded in the **Test Case Progress** area. Also, the **Status** of each test item (**Passed**, **Failed**, or **Pending**) is listed next to the **Manifest File** name.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

4. If desired, you can also enter comments in the **Comment** field on this dialog box.

Checking ODBC Data Sources

The **ODBC Data Sources** Deployment Test is run to verify ODBC data sources.



Task:

To check ODBC data sources:

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **ODBC Data Sources** from the View List. A list of data sources in that application appears in the lower portion of the **ODBC Data Sources View**. Only those data sources that belong to the current logged-in user are listed on the **ODBC Data Sources View**.
3. Select the Test Item you want to run and click **Run**. You can also select multiple Test Items to run, or click **Run All** to run all available Test Items.

For certain ODBC data sources, additional connection information is required for verification. When the tests are run in Full user interface mode, additional dialog boxes may be displayed during the test to take more input. However, when the tests are run in Silent user interface mode, these additional dialog boxes will not be displayed and results will be based on default information.

4. When testing is finished, results are recorded in the **Test Case Progress** area.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

Checking ODBC Drivers

The **ODBC Drivers** Deployment Test is run to verify ODBC Drivers.



Task:

To check ODBC drivers:

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. Expand the **Deployment Tests** node and select **ODBC Drivers** from the View List. A list of drivers in that application appears in the lower portion of the **ODBC Drivers View**. Only those drivers that belong to the current logged-in user are listed.
3. Select the Test Item you want to run and click **Run**. You can also select multiple Test Items to run, or click Run All to run all available Test Items.

For certain ODBC drivers, additional connection information is required for verification. When the tests are run in Full user interface mode, additional dialog boxes may be displayed during the test to take more input. However, when the tests are run in Silent user interface mode, these additional dialog boxes will not be displayed and results will be based on default information.

4. When testing is finished, results are recorded in the **Test Case Progress** area.



Note • When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

Specifying Exclusions for Deployment Testing

When a deployment test is run on a package, some of the tests related to Class IDs, Prog IDs, and Type Library IDs fail because they refer to components which belong to the operating system rather than the software which is being tested. These errors have no impact on the integrity of the software being tested, and cause confusion among some users testing the software. Users need to be able to prevent error messages caused by files that are not affecting the performance of the software package to be listed in the test results.

To prevent these operating systems errors from being reported, you can specify a list of files to be excluded when any of the Class ID, Prog ID, or Type Library ID Deployment tests are run. You can maintain a different list for each of these three Deployment Tests.



Note • After a Deployment Test has been run, the test results are listed in the [Class IDs View](#), [Prog IDs View](#), or [Type Libraries View](#). The items included in the exclusion lists are not shown in these views, but are still stored in the QualityMonitor Project File (.iqm). When this project file is opened again in QualityMonitor, the results are checked against the exclusion list before being displayed in the [Class IDs View](#), [Prog IDs View](#), or [Type Libraries View](#).

To Add a File to the Exclusion List

On the **Exclusions** tab of the [Options Dialog Box](#), you can manage all three exclusion lists. On the **Exclusions** tab, you can view the exclusion lists, and can add or remove entries from a list.

Using the Options Dialog Box



Task:

To exclude Class IDs, Prog IDs, or Type Library IDs using the Options dialog box:

1. From the QualityMonitor interface, select **Options** on the **Tools** menu. The **Options** dialog box opens.
2. Click the **Exclusions** tab. On the **Exclusions** tab, excluded items are listed for the selected **Exclusions list**.
3. From the **Exclusions list**, select the Deployment Test that you want to modify the exclusion list for: **Class ID**, **Prog ID**, or **Type Library**.
4. Click **Add**. The **Add Exclusions** dialog box opens.
5. Next to the **File Name** box, click **Browse** and select the **Application (.exe)**, **Application Extension (.dll)**, **Type Library (.tlb)**, or **ActiveX object (.ocx)** file that contains Class IDs, Prog IDs, or Type Library IDs that you want to exclude from the selected Deployment Test.

The Class IDs, Prog IDs, or Type Library IDs that are associated with the selected file are listed, displaying the **Identifier** and a **Description** of each.

6. Select the Class IDs, Prog IDs, or Type Library IDs that you want to exclude from the Deployment Test and click **OK**.

Directly From the Results Window

After a Deployment Test has been run and the test results are listed in the [Class IDs View](#), [Prog IDs View](#), or [Type Libraries View](#), you can add an item to the exclusion list directly by right-clicking on the item you want to exclude and choosing **Add to exclusion** from the context menu.

Status	Prog ID	Description	File Name
Passed	PSP7.Image	Paint Shop Pro 7 Image	C:\Program
Failed	PSP7.MultiImagePrint	Paint Shop Pro 7 MultiImage Printing File	C:\Program
Passed	FTI.Device.Digita.InfraredCtrl.1	FTI Device Digita Infrared Control	C:\Program
Passed	FTI.Device.Digita.SerialCtrl.1	FTI Device Digita Serial Control	C:\Program
Passed	FTI.Device.Digita.USBCtrl.1	FTI Device Digita USB Control	C:\Program
Passed	HOTLINK.HotLinkCtrl.1	HotLink Control	C:\Program
Failed	PSP7.BrowserFile	Paint Shop Pro 7 Browser Cache File	C:\Program
Failed	JMC.Docum	Microsoft Office Album	C:\Program
Failed	PSP7.WorkS	Workspace File	C:\Program
Failed	AnimationSh	Animation	C:\Program
Failed	AnimationSh	space	C:\Program
Passed	StdFont		C:\WINDOWS
Passed	OldFont		C:\WINDOWS
Passed	StdPicture		C:\WINDOWS
Passed	PSP6.Image	ge	C:\Program
Passed	PSP5.Image	ge	C:\Program

Figure 17-1: Adding a Prog ID to the Prog ID Exclusion List from the Test Results List

Deleting a Item from an Exclusion List

To delete an item from an exclusion list, perform the following steps.



Task: *To delete a Class IDs, Prog IDs, or Type Library IDs from an exclusion list:*

1. From the QualityMonitor interface, select **Options** on the **Tools** menu. The **Options** dialog box opens.
2. Click the **Exclusions** tab. On the **Exclusions** tab, excluded items are listed for the selected **Exclusions list**.
3. From the **Exclusions list**, select the Deployment Test that you want to modify the exclusion list for: **Class ID**, **Prog ID**, or **Type Library**.
4. Select the item that you want to delete from the list and click **Remove**. The item is removed from the list.

Selecting the Default Exclusion List

On the [General Tab](#) of the **Options** dialog box, you can select the default **Exclusion file** to use to filter the test results in the **Lockdown and Runtime Tests** views. By selecting an exclusion file from a shared location, multiple people can use the same error exclusion settings.

Lockdown and Runtime Testing

Lockdown and runtime tests are available through the [Lockdown and Runtime Tests View](#). You are provided with a list of available shortcuts in the package and all of the executables in the package. You can then launch via the shortcut or executable, and exercise functionality in the application. When you close the application, information about the executable is listed under the **Runtime Checks** node. This information, grouped into **Files**, **Registry Entries**, and **Folders** views, allows you to see failures in the application execution. These are potential issues with the application, and may or may not have any affect on the overall package integrity.

If you want to execute tests in the context of a different user (under a different user account), click **Run As** instead of **Run** to execute the test. You would then be prompted to enter a **User Name** and **Password**. For more information, see [Performing Lockdown and Runtime Tests Under a Different User Account](#).

This section includes the following topics:

- [Performing Lockdown and Runtime Tests](#)
- [Performing Lockdown and Runtime Tests Under a Different User Account](#)
- [Running Lockdown and Runtime Tests in Restricted Environments](#)
- [Performing Isolation Tests](#)
- [Filtering Results of Lockdown and Runtime Tests](#)



Note • Lockdown and runtime tests are only supported on Windows NT4, 2000, and XP.

Performing Lockdown and Runtime Tests

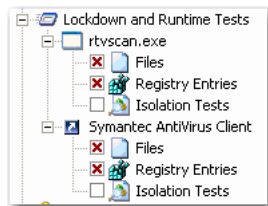
Lockdown and Runtime Tests allow you to see failures in the application execution. These are potential issues with the application, and may or may not have any affect on the overall package integrity.



Task: *To perform lockdown and runtime tests:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. From the View List, select **Lockdown and Runtime Tests**. The **Lockdown and Runtime Tests View** opens.
3. Select either the **Select a Shortcut** or **Select an Executable** option.
4. Select the shortcut or executable to run.
5. Click **Run**.
6. When the application launches, use the application in a normal way, performing various operations.
7. Exit the application.

The name of the executable or shortcut is now listed as a new node under the **Lockdown and Runtime Tests** node in the View List, and a subnode is listed for any access failures for **Files**, **Folders**, or **Registry Entries**.



8. Select the **Files**, **Folders**, and **Registry Entries** nodes. The right side of the **Lockdown and Runtime Tests View** displays a list of failed **Test Items** for each node. For information on specifying which errors are listed, see [Filtering Results of Lockdown and Runtime Tests](#).
9. To view the error message for a **Test Item**, right-click the Test Item and select **Test Item Information** on the context menu. The Test Item Information dialog box opens, listing an **Error Description** in the **Test Details** area.

Performing Lockdown and Runtime Tests Under a Different User Account

You can use the **Run As** feature to execute Lockdown and Runtime tests in the context of a different user. This allows you to validate an application in a locked-down environment without actually requiring a user to log-in with a different set of credentials. This will reduce the test cycle effort significantly.



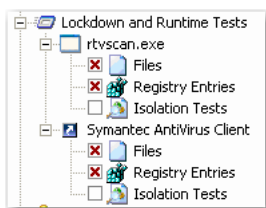
Task: *To perform tests under a different user account:*

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. From the View List, select **Lockdown and Runtime Tests**. The **Lockdown and Runtime Tests View** opens.
3. Select either the **Select a Shortcut** or **Select an Executable** option.
4. Select the shortcut or executable to run.
5. Click **Run As**.

You are then prompted to enter a **User Name** and **Password**.

6. Enter the User Name in the format of: `DOMAINNAME\UserName`. The default value is the current **User Name**.
7. When the application launches, use the application in a normal way, performing various operations.
8. Exit the application.

The name of the executable or shortcut is now listed as a new node under the **Lockdown and Runtime Tests** node in the View List, and a subnode is listed for any access failures for **Files**, **Folders**, or **Registry Entries**.



9. Select the **Files**, **Folders**, and **Registry Entries** nodes. The right side of the **Lockdown and Runtime Tests View** displays a list of failed **Test Items** for each node. For information on specifying which errors are listed, see [Filtering Results of Lockdown and Runtime Tests](#).
10. To view the error message for a **Test Item**, right-click the Test Item and select **Test Item Information** on the context menu. The Test Item Information dialog box opens, listing an **Error Description** in the **Test Details** area.



Note • **Run As** can also be selected using the **Shift+F5** shortcut, or by selecting **Run As** from the **Execute** menu.

Running Lockdown and Runtime Tests in Restricted Environments

When executing runtime tests in a locked-down environment, you may encounter an error such as:

Unable to monitor the application execution.

QualityMonitor needs to run under a user with Admin privileges when executing Lockdown and Runtime Tests.

To emulate lockdown environment under a restricted user, click **Run As** instead of **Run** to execute the test, and provide the **User Name** and **Password** of a locked down user. For more information, see [Performing Lockdown and Runtime Tests Under a Different User Account](#).

Performing Isolation Tests

You can run Isolation Tests to display the location of all portable executable (PE) files (d11/ocx/exe/t1b/o1b) that are launched from a process while performing a Lockdown and Runtime test. Viewing a listing of these portable executable file names and paths makes it easier for you to ensure that the application is fully isolated.

After you perform a Lockdown and Runtime Test for an executable (.exe) or a shortcut on the [Lockdown and Runtime Tests View](#), an additional node called **Isolation Tests** is added to the tree under the executable or shortcut node.

When you select this **Isolation Tests** node, the filenames of the portable executable files and their paths are listed. By default, the status of all these items is **Pending**. To ensure that all of the executables or shortcuts in this test case are isolated, go to the **Test Case Status** area of the view, and set the status of the entire test case to either **Pending**, **Passed**, or **Failed**.



Note • Note the following regarding Isolation testing:

- The Isolation Tests node will be added to the Lockdown and Runtime Tests tree only if the selected executable launches at least one portable executable file.
- You can specify when you would like the **Isolation Tests** node to appear in the Lockdown and Runtime Tests tree by selecting an option from the **Show Isolation Tests** list on the **General** tab of the [Options Dialog Box](#).
- QualityMonitor does not support isolation testing under Windows 2000.

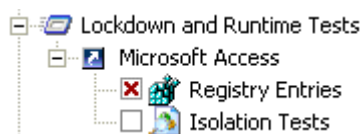


Task: To perform isolation tests:

1. Create or open a QualityMonitor project. The QualityMonitor **Product Information View** opens.
2. From the View List, select **Lockdown and Runtime Tests**. The **Lockdown and Runtime Tests View** opens.
3. Select either the **Select a Shortcut** or **Select an Executable** option.
4. Select the shortcut or executable to run.

5. Click **Run**.

After a Lockdown/Runtime test is performed for an executable (.exe) or a shortcut, an additional node called **Isolation Tests** is added to the tree under the executable or shortcut node.



6. Select the **Isolation Tests** node. The filenames of the portable executable files and their paths is displayed in the list control of the view. By default, the status of all of the items in this test case is **Pending**.
7. To ensure that all of the executables or shortcuts in this test case are isolated, go to the **Test Case Status** area of the view, and set the status of the entire test case to either **Pending**, **Passed**, or **Failed**.

Filtering Results of Lockdown and Runtime Tests

On the Lockdown and Runtime Tests [Files View](#), [Folders View](#), [Registry Entries View](#), and [Isolation Tests View](#) you can choose to filter the results that are listed.



Task:

To filter test results:

1. To filter the list by Test Item status, select an option from the **View these test items** list: **Passed**, **Failed**, **Pending**, or **All**.
2. To select errors to exclude from future Lockdown and Runtime result listings, perform the following steps:
 - a. Click the **Set Filter** button. The [Runtime Test Filters Dialog Box](#) opens, listing all errors that were generated during this test.
 - b. Select those errors that you want to exclude from future Lockdown and Runtime tests.

These settings are stored in the default exclusion list (the **Exclusion file** selected on the [General Tab](#) of the **Options** dialog box), but no changes are made to the Project file. This filter is based on the error code associated with an error, and these error codes are stored in the default exclusion list.
3. To filter the list by one type of error that was generated, make a selection from the **Having these errors** list. This list includes all the unique errors that were generated when this Test Case was executed (excluding the errors that were filtered out using the **Set Filters** function). To see all the errors, select **Show All**.

Using MSI Doctor to Verify Package Deployment Status

You can use QualityMonitor's MSI Doctor to verify if an MSI package is installed properly. This helps prevent users from seeing an auto-repair dialog box when they run the application. Auto repair messages are displayed by applications to attempt to reinstall missing/corrupted components.

By examining an application using QualityMonitor MSI Doctor, you can quickly identify any problems by checking the status of all products and features. Using MSI Doctor, you can:

- See the status of all components
- Verify if any files are missing or if any files do not match the version or size specified in the MSI file
- See the components status segregated by features
- Configure or reinstall features
- Reinstall components

To use MSI Doctor, select the **Deployment Status** node (📁) from the QualityMonitor View List to access the [Deployment Status View](#). Under the **Deployment Status** node, a tree view of the application's features are components are listed:

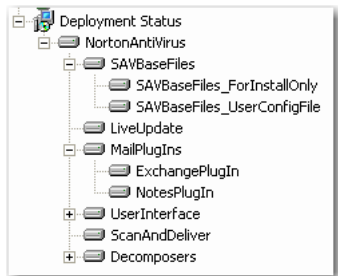




Figure 17-2: Deployment Status View: Features and Components

The following icons are displayed in the tree view and in the component list and indicate the feature or component's status:

Table 17-3 • Deployment Status View Icons

Icon	Name	Description
	Installed	Feature or Component is installed on the local machine.
	Uninstalled	Feature or Component is not installed on the local machine.
	Broken	Component is broken (a key file in the Component is missing) or Feature contains a broken Component.

Table 17-3 • Deployment Status View Icons (cont.)

Icon	Name	Description
	Run From Source	Feature or Component is configured to run from a source location (rather than being installed on the local machine).
	On Demand	Feature is configured to be installed when needed. Not applicable to Components.

Context Menu Functionality

The following table lists the functions available on the context menus for the **Deployment Status** icon, **Features**, and **Components**, and the dialog boxes that appear when those functions are selected:

Table 17-4 • Dialog Boxes Invoked from Deployment Status View Context Menu

Context Menu	Deployment Status Node	Feature Node	Component Node
Configure	Install or Configure Product Dialog Box	Install or Configure Feature Dialog Box	N/A
Re-install	Re-install Product/Feature Dialog Box	Re-install Product/Feature Dialog Box	Installation program is launched. (This option is only enabled if the selected Component is broken.)
Properties	Product Properties Dialog Box	Feature Properties Dialog Box	Component Properties Dialog Box

Using MSI Doctor, you can perform the following tasks:

- [View Product, Feature, or Component Deployment Status Properties](#)
- [Verify Product, Feature, or Component Data](#)
- [Install or Configure Products or Features](#)
- [Reinstall Features](#)

View Product, Feature, or Component Deployment Status Properties

To view the deployment status properties of a product, feature, or component, perform the following steps.



Task: *To view product, feature, or component Deployment Status properties:*

1. Launch QualityMonitor and open the package that you want to view the deployment status properties of. The QualityMonitor **Product Information View** opens.
2. Select one of the following:
 - **To view Product properties**—Select the **Deployment Status** icon from the View List.
 - **To view Feature properties**—Select a **Feature** icon under the **Deployment Status** icon on the View List.
 - **To view Component properties**—Select either the **Deployment Status** icon or a **Feature** icon from the View list, and then select a **Component** from the list on the right.
3. Select **Properties** from the context menu. The **Product Properties**, **Feature Properties**, or **Component Properties** dialog box appears, displaying property information for the selected Product, Feature, or Component.
4. On the Properties dialog boxes, you can also click the **Verify Data** button to verify if the files and registry information for the selected item are installed properly. See [Verify Product, Feature, or Component Data](#) for more information.
5. Click **OK** to exit the **Properties** dialog box.



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Verify Product, Feature, or Component Data



You can verify if the files and registry information for a Product, Feature, or Component are installed properly. Verification errors are displayed on the [Installed Data Dialog Box](#).




Task: *To verify product, feature, or component data:*

1. Launch QualityMonitor and open the package that you want to verify the files of. The QualityMonitor **Product Information View** opens.
2. Select one of the following:

- **To verify all of the files in all of the Features in the Product**, select **Deployment Status** from the View List.
 - **To verify only the files in the selected Feature**, select a **Feature** icon under **Deployment Status** on the View List.
 - **To verify only the files in a specific Component of a Feature**, select either the **Deployment Status** icon in the View List or a **Feature** icon under it, and then select a component from the list on the right.
3. Select **Properties** from the context menu. The **Product Properties**, **Feature Properties**, or **Component Properties** dialog box appears, displaying property information for the Product, Feature or Component.
 4. Click **Verify Data**. When you click **Verify Data**, QualityMonitor checks all of the files and registry entries included in the Product, Feature, or Component and then displays them on the **Files** and **Registry** tabs of the **Installed Data** dialog box.
 - On the **Files** tab, the following icons are used to identify verification errors:

Icon	Description
	File is missing.
	File has a different version or size than that specified in the Windows Installer package

- On the **Registry** tab, all registry entries for the selected item are listed. Registry data is verified by checking the existence of the registry key and the value name (if one exists). (The value data is not checked.). The  icon is used to indicate that a registry key or value name is incorrect or missing:
5. Click **Close** to exit the **Installed Data** dialog box, and click **OK** to exit the **Properties** dialog box.



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Install or Configure Products or Features

You can use **Configure** to install a product or feature that is not currently installed.




Task: **To install or configure products or features:**

1. Launch QualityMonitor and open the package that you want to install or configure. The QualityMonitor **Product Information View** opens.
2. Select one of the following:
 - **To install or configure the entire Product**, select **Deployment Status** from the View List.

- **To install or configure only the selected Feature**, select a **Feature** icon under **Deployment Status** on the View List.
- 3. Select **Configure** from the context menu. The **Install or Configure Product** or **Install or Configure Feature** dialog box appears, prompting you to select the installation location and the installation type (on the **Install or Configure Product** dialog box only).
- 4. Select one of the following options to specify installation location:
 - **Default**—Files will be installed to their default location.
 - **Local**—Files will be installed on the local machine.
 - **Source**—Files will be run from the installation source.
 - **On Demand**—Files will be installed when needed.
- 5. (Product only) Select one of the following options to specify installation type:
 - **Minimum**—Only the essential features will be installed.
 - **Typical**—Most commonly used features will be installed.
 - **Complete**—All of the program's features will be installed.
- 6. Click **OK**. The Product or Feature is installed, per the options you specified.

Reinstall Features

When a Feature is broken (identified by the  icon), you can fix it by re-installing the entire Product or just re-installing the broken Feature.




Task: *To reinstall features:*

1. Launch QualityMonitor and open the package that you want to reinstall. The QualityMonitor **Product Information View** opens.
2. Select one of the following:
 - **To reinstall the entire Product**, select **Deployment Status** from the View List.
 - **To reinstall only the selected Feature**, select a **Feature** icon under **Deployment Status** on the View List.
3. Select **Re-install** from the context menu. The **Re-install Product/Feature** dialog box appears, prompting you to select a reinstall mode.
4. Select one of the following reinstall modes:
 - Repair all detected reinstall problems
 - Reinstall only if file is missing
 - Force all files to be reinstalled

- Reinstall if file is missing, or an older version exists
 - Reinstall if file is missing, or an older or equal version exists
 - Reinstall if existing file has different version
 - Verify that required user registry entries are present
 - Verify that required local machine registry entries are present
 - Recreate all shortcuts
5. Click **OK**. The Product or Feature is re-installed, per the option you specified.

Reinstall Components

When a Component is broken (identified by the  icon), you can fix it by re-installing the broken Component.



Task: *To reinstall components:*

1. Launch QualityMonitor and open a package. The QualityMonitor **Product Information View** opens.
2. Select **Deployment Status** from the View List or select a **Feature** icon under **Deployment Status** on the View List. All components associated with the selected feature(s) are listed.
3. Select the component that you want to re-install and select **Re-install** from the context menu. The component is automatically reinstalled.

Creating Custom Test Cases

QualityMonitor supports adding additional, custom Test Cases to projects—based on your business needs.

You can define a template that includes information that is re-used when defining Test Cases. Then, when creating a new user-defined Test Case, you can load this template file and have the components of the new Test Case pre-populated with the information saved in the template. A single template file can load multiple user defined Test Cases.

Adding User-Defined Test Cases



Task: *To add a user-defined case:*

1. Launch QualityMonitor and open the package that you want to create a custom Test Case for. The QualityMonitor **Product Information View** opens.
2. Right-click on the **User-Defined Tests** node and select **Add Test Case** from the context menu.

A new Test Case appears below the **User Defined Tests** node, and you are prompted to enter a name.

3. Name the Test Case appropriately.
4. Select the new Test Case to open the **Test Case View**.
5. Under **Test Case Status**, specify the status for this Test Case: **Pending**, **Passed**, or **Failed**.
6. Click **Browse** and select an executable to associate with this Test Case, if necessary.
7. In the **Instructions** text box, enter any necessary comments.

For example, you may want to create a custom Test Case to ensure that a specific database is updated properly after running an application.

8. Your entries are automatically saved in the new Test Case.

Creating and Using Test Case Templates



Task: *To create and use Test Case templates:*

1. Create a new Test Case as described in [Adding User-Defined Test Cases](#).
2. To save the Test Case as a template to re-use when creating new Test Cases, perform the following steps:
 - a. Select this Test Case node in the **User Defined Tests** tree and select **Save as template** from the context menu.
 - b. Specify a name and a location for the Test Case template and select **Save**. The template is saved in .xml format in the location you specify.
3. To add this Test Case to an existing template, perform the following steps:
 - a. Select this Test Case node in the **User Defined Tests** tree and select **Add to template** from the context menu. You are prompted to select the template file that you want to add this Test Case to.
 - b. Select the template that you want to add this Test Case to and select **Save**.
4. To create a new Test Case based upon a template, perform the following steps:
 - a. Select the **User Defined Tests** node and select **Load template** from the context menu. You are prompted to select the template that you want to use.
 - b. Select a template and click **Open**. All of the Test Cases that were saved in the template are now listed in the **User Defined Tests** tree.

You can specify that you want a specific template file automatically loaded each time a QualityMonitor project is opened. To do this, select a **Template file** on the [General Tab](#) of the **Options** dialog box, and also select the **Load Templates on Project Open** option.

Renaming User-Defined Test Cases



Task: *To rename a user-defined Test Case:*

1. Under the **User-Defined Tests** node, select the Test Case that you want to rename, and select **Rename** from the context menu.
2. Provide a new name for the Test Case.

Test Reports

QualityMonitor allows you to create an HTML test report for the current project. This can be done by selecting **Generate Report** from the **File** menu and providing the name and location for the report. The report will then automatically open in your default browser.



Note • *If an error message occurs when generating the report, it may be because msxml4.dll or isqm.xlst is not in the same directory as the QualityMonitor executable (isqm.exe). These files must be present to create the report.*

Running QualityMonitor from the Command Line

QualityMonitor can be run from the command line by using `isqm.exe`. It can accept the following parameters:

Table 17-5 • QualityMonitor Command Line Parameters

Syntax	Options	Description
-c <MSI Product Code>		Launch QualityMonitor by opening the MSI product specified by the product code. For example: <code>isqm.exe -c {BDC62375-07B4-4CBD-9991-4C25C24F3071}</code> <code>isqm.exe -c {BDC62375-07B4-4CBD-9991-4C25C24F3071} -sb -f <c:\projectfile.iqm></code>
	-sn	Run QualityMonitor silently without any user interaction and no progress display.
	-sb	Run QualityMonitor silently with a progress display. QualityMonitor displays the test names as they are executed and provides an option for the user to cancel.
	-r <Report File>	Generates a report file <c:\report.htm> with the test results. Works only when using either <code>-sn</code> or <code>-sb</code> .
	-f <Project File>	Save test results in this file. This is necessary when using either <code>-sn</code> or <code>-sb</code> . If this file does not exist, it will be created and then results will be saved.
-f <Project File>		Launch QualityMonitor with this Project File. For example: <code>isqm.exe -f c:\mydocuments\mytesting\tesresults.iqm</code> <code>isqm.exe -f c:\mydocuments\mytesting\tesresults.iqm -sn</code>
	-sn	Run QualityMonitor silently without any user interaction and no progress display.
	-sb	Run QualityMonitor silently with a progress display. QualityMonitor displays the test names as they are executed and provides an option for the user to cancel.
	-r <Report File>	Generates a report file <c:\report.htm> with the test results. Works only when using either <code>-sn</code> or <code>-sb</code> .
-h or -?		Help



Note • When using any `-sn` or `-sb` command line options, you can specify the target product using the existing options `/c` or `/f`.

- If you use the `/f` option to specify the product, the input file will be modified with the test results.
- If you use `/c` to specify the target product, `/f` options must be used to specify the project file path which will have the test results.
- If both `/c` and `/f` parameters are specified, then QualityMonitor gives preference to `/c` and operates with the product code specified by `/c`.

QualityMonitor Reference

Topics contained in this section provide detailed reference on each user interface element, dialog box, or view in QualityMonitor. This is the same documentation displayed when you click F1 from the QualityMonitor interface. Topics are organized as follows:

- [Menus and Toolbar](#)
- [QualityMonitor Interface](#)
- [Dialog Boxes](#)
- [Views](#)

Menus and Toolbar

The following table provides a description of each of QualityMonitor's menu commands and toolbar buttons:

Table 17-6 • QualityMonitor Menus and Toolbar Options



Menu	Command	Toolba r Button	Keyboar d Shortcut	Description
File	Open		Ctrl+O	Allows you to open an existing QualityMonitor project file (.iqm) or create a new one based on an installed MSI-based application.
File	Close			Closes the current project.
File	Save		Ctrl+S	Saves the current project.
File	Save As			Saves the current project using the name and location you specify.

Table 17-6 • QualityMonitor Menus and Toolbar Options (cont.)




Menu	Command	Toolbar Button	Keyboard Shortcut	Description
File	Generate Report			Creates an HTML test report for the current project. The report will automatically open in your default browser.
File	1,2,3,4			Allows you to open the four most recently accessed QualityMonitor projects.
File	Exit			Exits QualityMonitor.
View	Toolbar			Toggles display of the toolbar.
View	Status Bar			Toggles display of the status bar.
Test Case	Add Test Case			Adds a custom Test Case beneath the Additional Tests view in the View List.
Execute	All Deployment Tests		Ctrl+F5 or Alt+E+D	Runs all the deployment tests in the current project.
Execute	Run		F5 or Alt+E+R	Runs the selected deployment, lockdown and runtime, or user defined test.
Execute	Run As		Shift+F5	Runs the selected lockdown and runtime test in the context of a different user. You are then prompted to enter a User Name and Password.
Tools	Options			Displays the Options dialog box.
Help	Contents			Launches the Help Library, displaying the Contents tab.
Help	Index			Launches the Help Library, displaying the Index tab.
Help	Search			Launches the Help Library, displaying the Search tab.
Help	Support Central			Accesses AdminStudio Support Central on the Web.
Help	Web Community			Accesses the AdminStudio Community on the InstallShield Web site.
Help	ReadMe			Displays the AdminStudio ReadMe file.

Table 17-6 • QualityMonitor Menus and Toolbar Options (cont.)

Menu	Command	Toolbar Button	Keyboard Shortcut	Description
Help	Feedback			Accesses the AdminStudio feedback form on the InstallShield Web site.
Help	AdminStudio on the Web			Accesses the AdminStudio Web site.
Help	About QualityMonitor			Displays the About dialog box.

QualityMonitor Interface

The QualityMonitor interface is divided into two main areas. The View List, which appears at the left side of the screen, provides a visual representation of the QualityMonitor project file. It provides easy access to individual views and Test Cases. The right side of the interface changes depending on the view or Test Case selected.

Dialog Boxes

The following dialog boxes are accessible from within QualityMonitor:

- [About QualityMonitor Dialog Box](#)
- [Add Exclusions Dialog Box](#)
- [Component Properties Dialog Box](#)
- [Feature Properties Dialog Box](#)
- [Install or Configure Feature Dialog Box](#)
- [Install or Configure Product Dialog Box](#)
- [Installed Data Dialog Box](#)
- [Open QualityMonitor Project Dialog Box](#)
- [Options Dialog Box](#)
- [Product Properties Dialog Box](#)
- [Re-install Product/Feature Dialog Box](#)
- [Test Item Information Dialog Box](#)
- [Test Progress Dialog Box](#)
- [Test Result Dialog Box](#)

About QualityMonitor Dialog Box

The About InstallShield QualityMonitor dialog box, available by selecting About from the Help menu, displays information about QualityMonitor and AdminStudio, including version and serial number information.

Add Exclusions Dialog Box

On the Add Exclusions dialog box, which is opened by clicking **Add** on the [Exclusions Tab](#) of the Options dialog box, you can select a Class ID, Prog ID, or Type Library ID to exclude from a Deployment Test. See [Specifying Exclusions for Deployment Testing](#).

The following options are included:

Table 17-7 • Add Exclusions Dialog Box Options

Options	Description
File Name	Click Browse and select the Application (.exe) , Application Extension (.dll) , Type Library (.tlb) , or ActiveX object (.oxc) file that contains Class IDs, Prog IDs, or Type Library IDs that you want to exclude from the selected Deployment Test.
Identifier	Identifies the Class IDs, Prog IDs, or Type Library IDs that are associated with the selected file.
Description	Description of the Class IDs, Prog IDs, or Type Library IDs that are associated with the selected file.

Component Properties Dialog Box

The Component Properties dialog box is displayed when you select a Component in the Component list on the right side of the [Deployment Status View](#) and select Properties from the context menu.

The following information is listed:

Table 17-8 • Component Properties Dialog Box Options

Options	Description
Name	Name of the selected Component.
GUID	Number which uniquely identifies this Component.
Status	Status of Component, such as Installed on Local Drive.
Location	Location on server where Component is installed.

Table 17-8 • Component Properties Dialog Box Options (cont.)

Options	Description
Verify Data	<p>Click to verify the following for the selected Component:</p> <ul style="list-style-type: none">• Files—QualityMonitor verifies the existence of files in the specified location, and compares the file size specified in the Windows Installer .msi package to that of the file on the system. Missing or modified files are identified.• Registry—QualityMonitor verifies the Registry data by checking the existence of the registry key and the value name (if one exists). Only the registry keys and value names are verified; the values themselves are not verified. Missing or incorrect registry keys are identified. <p>The Files and Registry information is listed on the Installed Data Dialog Box.</p>



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Feature Properties Dialog Box

The Feature Properties dialog box is displayed when you select a Feature under the **Deployment Status** node and then select **Properties** from the context menu.

This dialog box contains the following options:

Table 17-9 • Feature Properties Dialog Box Options

Options	Description
Name	Name of selected Feature.
Title	Title of selected Feature.
Parent	Parent Feature of this Feature (if one exists).
Description	Description of this Feature.
Last Used	Date this Feature was last used.
Usage Count	Number of times this Feature has been used.

Table 17-9 • Feature Properties Dialog Box Options (cont.)

Options	Description
Verify Data	<p>Click to verify the following for the selected Feature:</p> <ul style="list-style-type: none"> • Files—QualityMonitor verifies the existence of files in the specified location, and compares the file size specified in the Windows Installer .msi package to that of the file on the system. Missing or modified files are identified. • Registry—QualityMonitor verifies the Registry data by checking the existence of the registry key and the value name (if one exists). Only the registry keys and value names are verified; the values themselves are not verified. Missing or incorrect registry keys are identified. <p>The Files and Registry information is listed on the Installed Data Dialog Box.</p>



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Install or Configure Feature Dialog Box

The Install or Configure Feature dialog box is displayed when you select a Feature under the Deployment Status node and then select Configure from the context menu.

If you select an option on this dialog box and click OK, QualityMonitor will attempt to install or configure the selected Feature to the settings you specify. Select one of the following options:

- **Default**—Files will be installed to their default location.
- **Local**—Files will be installed on the local machine.
- **Source**—Files will be run from the installation source.
- **On Demand**—Files will be installed when needed.



Note • To complete this operation, you may need the source from which the selected Feature was installed.

Install or Configure Product Dialog Box

The **Install or Configure Product** dialog box is displayed when you select the **Deployment Status** node and then select **Configure** from the context menu. If you select an option on this dialog box and click **OK**, QualityMonitor will attempt to install or configure the Product to the settings you specify.

Installation Location

Select one of the following options:

- **Default**—Files will be installed to their default location.
- **Local**—Files will be installed on the local machine.
- **Source**—Files will be run from the installation source.
- **On Demand**—Files will be installed when needed.



Note • To complete this operation, you may need the source from which the selected feature was installed.

Installation Type

Select one of the following options:

- **Minimum**—Only the essential Features will be installed.
- **Typical**—Most commonly used Features will be installed.
- **Complete**—All of the program's Features will be installed.

Installed Data Dialog Box

The Installed Data dialog box appears when you are using MSI Doctor to verify package deployment status, and you perform the following steps:





Task: **To view the Installed Data dialog box:**

1. Go to the [Deployment Status View](#) and select the **Deployment Status** node, one of the Features listed under it, or a component listed on the right, and then
2. Select **Properties** from the context menu to display the [Product Properties Dialog Box](#), [Feature Properties Dialog Box](#), or [Component Properties Dialog Box](#), and then
3. Click **Verify Data**. QualityMonitor then checks all of the files and registry entries included in the selected Product, Feature, or Component and then displays them on the **Installed Data** dialog box.

Files Tab


Files are verified by checking their existence in the specified location, and comparing the file size specified in the .Windows Installer package to that of the file on the system. On the **Files** tab, the following icons are used to identify verification errors:

Table 17-10 • Verification Error Icons

Icon	Description
	File is missing.
	File has a different version or size than that specified in the Windows Installer package

If you double-click on an item listed on the **Files** tab, the Windows Explorer opens to the folder containing the selected file.

Registry Tab

The **Registry** tab lists all of the registry keys and value names in the selected Product, Feature, or Component. Registry data is verified by checking the existence of the registry key and the value name (if one exists). The  icon is used to indicate that a registry key or value name is incorrect or missing.



Note • Only the registry keys and value names are verified; the values themselves are not verified.



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Open QualityMonitor Project Dialog Box

The Open QualityMonitor Project dialog box opens when you select to open a QualityMonitor project from the Welcome page, or when you either select **Open** from the **File** menu or click the **Open** button on the toolbar.

From this dialog box, you can either select to open an existing QualityMonitor file (and subsequently enter or browse to it), or select to create a project file based on an installed MSI-based application.


Options Dialog Box

The Options dialog box, available by selecting **Options** from the **Tools** menu, has options on two tabs: [General Tab](#) and [Exclusions Tab](#).

General Tab

The following options are included on the **General** tab:

Table 17-11 • Options Dialog Box—General Tab Options

Options	Description
Update the test case status automatically after executing test items	Select this option if you want to automatically update test case status after executing test items. If you do not use this functionality, the View List will not automatically update after a test item has been executed.
Show Isolation Tests	<p>Use this option to choose when you would like to Show Isolation Tests after performing a Lockdown and Runtime test. You have the following options:</p> <ul style="list-style-type: none">• Always—Show this view for all the executables run, irrespective of the presence of records in the IsolatedComponent table and in MsiAssembly SXS records.• Never—This view will not be shown irrespective of the data.• Only if the Application is Isolated—Show this view only if the MSI Package has either IsolatedComponent records or MSIAssembly SXS records.• Only if the Running Operating System supports Isolation  <p>Note • AdminStudio always stores the information in the project but, the UI selection you make here determines whether this view will be populated.</p>
Template file	Select a User Defined Tests template file. If you also select the Load Templates on Project Open open option, all of the Test Cases in the selected Template file will be automatically loaded when a QualityMonitor project is open.
Exclusion file	Select an exclusion file to use to filter the test results in the Lockdown and Runtime views. By selecting an exclusion file from a shared location, multiple people can use the same error exclusion settings.

Exclusions Tab

When a Deployment Test is run on a package, some of the tests related to Class IDs, Prog IDs, and Type Library IDs fail because they refer to components which belong to the operating system rather than the software which is being tested. These errors have no impact on the integrity of the software being tested, and cause confusion among some users testing the software. Users need to be able to prevent error messages caused by files that are not affecting the performance of the software package to be listed in the test results.

On the **Exclusions** tab, to prevent these operating systems errors from being reported, you can specify a list of files to be excluded when any of the Class ID, Prog ID, or Type Library ID Deployment Tests are run. You can maintain a different list for each of these three Deployment Tests.



Note • After a Deployment Test has been run, the test results are listed in the [Class IDs View](#), [Prog IDs View](#), or [Type Libraries View](#). The items included in the exclusion lists are not shown in these views, but are still stored in the QualityMonitor Project File (.iqm). When this project file is opened again in QualityMonitor, the results are checked against the exclusion list before being displayed in the [Class IDs View](#), [Prog IDs View](#), or [Type Libraries View](#).

The following options are included on the **Exclusions** tab:

Table 17-12 • Options Dialog Box—Exclusion Tab Options

Options	Description
Exclusion list	Select the type of Deployment Test that you want to modify the exclusion list for: <ul style="list-style-type: none"> • Class ID • Prog ID • Type Libraries
Listing	Listing of the Class IDs, Prog IDs, or Type Libraries that you have chosen to exclude from the selected Deployment Test. The following information is included: <ul style="list-style-type: none"> • Identifier—Number identifying the Class ID, Prog ID, or Type Library. • File Name—Name of file that contains the Class ID, Prog ID, or Type Library. • Status—The exclusion status is either Active or Inactive. Only Active exclusions are excluded from the test results. Inactive exclusions are neglected/omitted from the exclusion process. The only way to change the status from Active to Inactive (or vice versa) is to manually edit the exclusions file. • Type—The exclusion type is either User or System. Exclusions added by the user are of User type. Users can delete only User type exclusion entries.
Add	Click to open the Add Exclusions Dialog Box , where you can select a Class ID, Prog ID, or Type Library to exclude from Deployment Tests.
Remove	Click to delete the selected Class ID, Prog ID, or Type Library from the exclusion list.

Product Properties Dialog Box

The Product Properties dialog box is displayed when you select the **Deployment Status** node and then select **Properties** from the context menu.

This dialog box contains the following options:

Table 17-13 • Product Properties Dialog Box Options

Options	Description
Version	Product version.
Publisher	Manufacturer of Product.
Product Code	Number which uniquely identifies this Product.
Local	Directory on local machine where this MSI file is located.
Registered to	Registered user of Product.
Product ID Status	Installation status of this Product, such as “The product is installed for the current user.”
Help Link	Main help link for the Product.
Installed on	Date Product was installed.
Installed from	Location where Product was installed from.
Verify Data	<p>Click to verify the following for the selected Product:</p> <ul style="list-style-type: none">• Files—QualityMonitor verifies the existence of files in the specified location, and compares the file size specified in the Windows Installer .msi package to that of the file on the system. Missing or modified files are identified.• Registry—QualityMonitor verifies the Registry data by checking the existence of the registry key and the value name (if one exists). Only the registry keys and value names are verified; the values themselves are not verified. Missing or incorrect registry keys are identified. <p>The Files and Registry information is listed on the Installed Data Dialog Box.</p>



Note • To save all the deployment status information in the QualityMonitor project file (.iqm), select the **Save deployment status information when saving project** option on the [Options Dialog Box](#).

Re-install Product/Feature Dialog Box

The Re-install Product/Feature dialog box is displayed when you select a Feature under the **Deployment Status** node or you select the **Deployment Status** node and then select **Re-install** from the context menu.

If you select an option on this dialog box and click **OK**, QualityMonitor will attempt to reinstall the selected Feature(s) to the settings you specify. Select one option from the **Select Reinstall Mode** or **Additional Reinstall Modes** property:

Table 17-14 • Re-install Product/Feature Dialog Box Options

Option	Description
Select Reinstall Mode	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Repair all detected reinstall problems • Reinstall only if file is missing • Force all files to be reinstalled
Additional Reinstall Mode	<p>Select one of the following options:</p> <ul style="list-style-type: none"> • Reinstall if file is missing, or an older version exists • Reinstall if file is missing, or an older or equal version exists • Reinstall if existing file has different version • Verify that required user registry entries are present • Verify that required local machine registry entries are present • Recreate all shortcuts

Runtime Test Filters Dialog Box

The Runtime Test Filters Dialog box opens when you click the **Set Filters** button on one of the Lockdown and Runtime Test views: [Files View](#), [Folders View](#), [Registry Entries View](#), or [Isolation Tests View](#).

All of the errors that were generated for that Lockdown and Runtime Test Case are listed. If you want to exclude specific errors from future Lockdown and Runtime tests, select those errors and click **OK**.

These settings are stored in the default exclusion list (the **Exclusion file** selected on the [General Tab](#) of the **Options** dialog box), but no changes are made to the Project file.

Test Item Information Dialog Box

The Test Item Information dialog box is displayed when you right-click on a Test Item and select **Test Item Information** from the context menu. The following information is included:

Table 17-15 • Test Item Information Dialog Box Options

Option	Description
Test Item	The file name and path of the selected Test Item.
Status	The status of the selected Test Item: Passed , Failed , or Pending .
Comments	Any comments that were entered to document this Test Item.
Test Details	Specific test details that can include the error message associated with the selected Test Item that was generated during testing. These error messages can help you diagnose issues with the package.

Test Progress Dialog Box

The Test Progress dialog box opens when you execute Test Items. Note the following:

- If you are performing Test Cases which have automatic execution (such as Type Libraries, Prog IDs, Services, or Class IDs), this dialog box opens briefly and automatically closes when execution is complete.
- For non-automatic Test Cases (Help Files, File Associations, and Shortcuts), this dialog box opens for each Test Item selected, allowing you to run the test and perform any necessary manual actions.
- Following execution of Test Items, the [Test Result Dialog Box](#) appears.

Test Result Dialog Box

The Test Result dialog box opens following execution of each Test Case requiring manual operations (Help Files, File Associations, and Shortcuts).

You can enter **Comments** about the execution of the functionality, and click **Yes** or **No** depending on whether the Test Item passed.

Views

The following views are available in QualityMonitor:

- [Welcome to QualityMonitor View](#)
- [Product Information View](#)
- [Test Cycle Summary View](#)

- Deployment Tests View
 - Class IDs View
 - File Associations View
 - Help Files View
 - Prog IDs View
 - Shortcuts View
 - Type Libraries View
 - ODBC Data Sources View
 - ODBC Drivers View
 - Services View
- Lockdown and Runtime Tests View
 - Runtime Execution Details View
 - Files View
 - Folders View
 - Registry Entries View
 - Isolation Tests View
- User-Defined Tests View
 - Test Case View
- Deployment Status View

Welcome to QualityMonitor View

The Welcome to QualityMonitor View is the view that is displayed before a project is created or opened. From this view you can choose to create a new QualityMonitor project, browse to an existing project, or open the most recently used project.

This view also lists the three major steps involved in using QualityMonitor to ensure package quality:

- **Open project**—Create or open a QualityMonitor project.
- **Run tests**—Run deployment tests, runtime and lockdown tests, or user-defined custom tests.
- **Analyze results**—Analyze the results of the tests.

Product Information View

The Product Information view displays information about the package you are testing in QualityMonitor.

The following data is displayed:

Table 17-16 • Product Information View Options

Option	Description
Application Name	The name of the application.
Author	The person or company who created the application.
Product Code	The package's product code.
Package Code	The package's package code.
Installed On	The date when the package was installed on the system.
Version	The package's version.



Note • In the View List, this view is titled with the product name.

Test Cycle Summary View

The Test Cycle Summary view provides statistics on the number of Test Cases and Test Items in the QualityMonitor project, and the ratio of cases and items passed, failed, or pending. If you add additional Test Cases, or perform runtime checking, the number of Test Cases and items will increase.

Deployment Tests View

The Deployment Tests View provides a summary of all Deployment Tests.

Deployment tests help you with up to several Test Cases to run on your Windows Installer-based application. Tests are only available if the application has the corresponding associated data (for example, if there are no shortcuts, you cannot run the Shortcuts Test Case).

QualityMonitor includes the following deployment tests:

- [Class IDs View](#)
- [File Associations View](#)
- [Help Files View](#)
- [Prog IDs View](#)

- [Shortcuts View](#)
- [Type Libraries View](#)
- [ODBC Data Sources View](#)
- [ODBC Drivers View](#)
- [Services View](#)

Automatically Running All Deployment Tests Silently

You can choose to run all deployment tests silently (without prompting for user input) using either the Interface or the command line.

From the Interface

You can choose to run all deployment tests silently (without prompting for user input) by making a selection in the QualityMonitor interface.



Task: *To run all deployment tests silently from the Interface, do one of the following:*

1. On the QualityMonitor **Product Information View**, select the **Deployment Tests** root node and then do one of the following:
 - Click the **Execute All Deployment Tests** button.
 - Select **All Deployment Tests** from the **Execute** menu.
 - Click the **Execute All Deployment Tests** toolbar button:



When you select one of these options, a dialog box with a progress bar and an option to cancel will be displayed.

From the Command Line

You can also run all deployment tests silently by entering a command in the command line. See [Running QualityMonitor from the Command Line](#) for more information.

Class IDs View

The **Class IDs** Deployment Test is run to determine if the Class ID COM objects can be instantiated programatically.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

File Associations View

The **File Associations** Deployment Test is run to determine if all file extensions have been installed and associated correctly. This involves launching a file with this extension, and determining if the correct application was used.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Help Files View

The **Help Files** Deployment Test is run to determine if the help files are installed and can be launched correctly.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Prog IDs View

The **Prog IDs** Deployment Test is run to ensure that the Prog IDs COM objects can be instantiated programatically.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Shortcuts View

The **Shortcuts** Deployment Test is run to determine if each shortcut is installed and if it successfully launches the shortcut target.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Type Libraries View

The **Type Libraries** Deployment Test is run to determine if the Type Libraries COM objects can be instantiated programatically. COM data is tested silently, returning results in the **Test Case Progress** area and the queue.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Manifests View

The **Manifests** Deployment Test is run to test the manifests and assemblies used to isolate a Windows Installer package.

The Manifests Deployment Test tests information from the `MsiAssembly` and `MsiAssemblyName` tables. QualityMonitor reads through the manifest/assembly files and performs the baseline Class IDs, Prog IDs, or Type Libraries testing for each entry in the files.

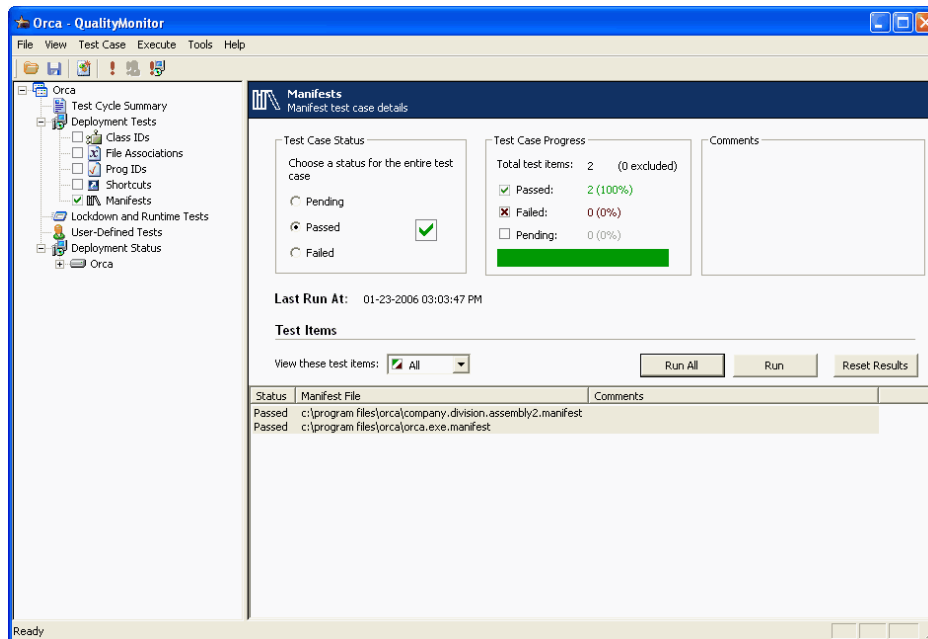


Figure 17-3: QualityMonitor Manifests View

Right-click on the **Test Item** you want to run and select **Run from the context menu**. You can also use the Shift or Ctrl keys to select multiple Test Items to run, or click **Run All** to run all available Test Items.

When testing is finished, results are recorded in the **Test Case Progress** area. Also, the **Status** of each test item (**Passed**, **Failed**, or **Pending**) is listed next to the **Manifest File** name.

When a Test Item is failed, you can view details about it, including the error message associated with it on the **Test Item Information** dialog box. To access this dialog box select the Test Item and then select **Test Item Information** from the context menu.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments. If desired, you can also enter comments in the **Comment** field on this view.

ODBC Data Sources View

The **ODBC Data Sources** Deployment Test is run to verify the ODBC data sources.

At the bottom of the view, you can see all ODBC DSNs associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.



Note • On the ODBC Data Sources View, only those data sources that belong to the current logged-in user are listed.

- For certain ODBC data sources, additional connection information is required for verification. When the tests are run in Full user interface mode, additional dialog boxes may be displayed during the test to prompt for more input. However, when the tests are run in Silent user interface mode, these additional dialog boxes will not be displayed and results will be based on default information.

ODBC Drivers View

The **ODBC Drivers** Deployment Test is run to verify ODBC drivers.

At the bottom of the view, you can see all ODBC drivers associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.



Note • On the ODBC Drivers View, only those drivers that belong to the current logged-in user are listed.

- For certain ODBC drivers, additional connection information is required for verification. When the tests are run in Full user interface mode, additional dialog boxes may be displayed during the test to take more input. However, when the tests are run in Silent user interface mode, these additional dialog boxes will not be displayed and results will be based on default information.

Services View

The **Services** Deployment Test is run to determine if all NT Services have been installed correctly. This is done by opening the Services Manager to determine if the Service exists on the target machine.

At the bottom of the view, you can see all Test Items associated with the Test Case, and can run these items either individually or simultaneously. You can also view individual Test Item details. At the top of the view, you can see and set the status of the entire Test Case, see Test Case progress, filter Test Item data, clear the results, or add comments.

Lockdown and Runtime Tests View

From the **Lockdown and Runtime Tests View**, you can select whether you want to perform runtime checking using a shortcut or an executable in the installed package.

You can select an item in the associated list and click **Run** to launch the application. After exercising the application's functionality and closing it, additional views will appear associated with the executable. These views initially display test items that failed during application operation, and are grouped into:

- [Runtime Execution Details View](#)
- [Files View](#)
- [Folders View](#)
- [Registry Entries View](#)
- [Isolation Tests View](#)

If you want to execute tests in the context of a different user (under a different user account), click **Run As**. For more information, see [Performing Lockdown and Runtime Tests Under a Different User Account](#).



Caution • Lockdown and runtime checks cannot be performed on Windows 9x-based systems.

Runtime Execution Details View

When a Lockdown and Runtime test is run on a package executable or shortcut, a new node with the name of that executable or shortcut is listed under the **Lockdown and Runtime Test Node**. When you select this executable or shortcut node, the Runtime Execution Details View opens, listing a summary of the execution of the Lockdown and Runtime tests.

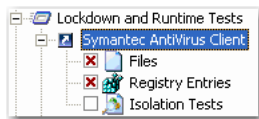


Figure 17-4: Shortcut Node under the Lockdown and Runtime Tests Node

On the Runtime Execution Details View, the following information is included:

Table 17-17 • Lockdown and Runtime Tests View / Runtime Execution Details View Options

Option	Description
Progress	<p>The Progress area includes the following information:</p> <ul style="list-style-type: none">• Total test cases—Number of test cases (Files, Registry Entries, Folders, Isolation Tests for the selected executable or shortcut) that generated failures plus those test cases that have not yet been completed.• Passed, Failed, Pending—Percentage of total test cases that passed the test, failed the test, or have not yet been executed. <p>If an executable or shortcut was run without any failures, the Progress area is not displayed.</p>
Last Run At	<p>Date and time that the last Lockdown and Runtime test was performed, and the name of the user who performed that test.</p>

Files View

When a Lockdown and Runtime test is run on a package executable or shortcut, a new node with the name of that executable or shortcut is listed under the **Lockdown and Runtime Test Node**. Under this executable or shortcut node, nodes for each Test Case that generated failures or has not yet been executed are listed. If any file errors were generated during this test, then the **Files** node appears.

When the **Files** node is selected, the Files View opens and includes the following information:]

Table 17-18 • Lockdown and Runtime Tests View / Files View Options

Option	Description
Test Case Status	<p>The Test Case Status area displays the state for the entire Test Case.</p> <ul style="list-style-type: none">• If failures were generated when this Test Case was executed, QualityMonitor sets the status to Failed.• If one of the Test Cases has not yet completed, QualityMonitor sets the status to Pending.• If no failures were generated, QualityMonitor sets the status to Passed. <p>Depending on your business practices and standards, you may want to override the status of a Test Case from its current state. In this instance, you would manually select another status. In most cases, this will be setting a Test Case which QualityMonitor has marked as Failed (because one or more individual Test Items have failed) to Passed.</p>

Table 17-18 • Lockdown and Runtime Tests View / Files View Options (cont.)

Option	Description
Test Case Progress	<p>The Test Case Progress area includes the following information:</p> <ul style="list-style-type: none"> • Total test items—Number of files that were tested when the shortcut or executable was run. The number of files that were excluded is also listed. • Passed, Failed, Pending—Percentage of total test items that passed the test, failed the test, or have not yet been executed.
Comments	Enter comments to document any special considerations or facts regarding this Test Case.
Test Items List	This lists all of the files that were executed when this Lockdown and Runtime Test was executed.
View these test items	Select one of the following to filter the file listing: All , Passed , Failed , or Pending .
Set Filter	<p>Click on this button to open the Runtime Test Filters Dialog Box, which lists all errors that were generated during this test. You can then choose to select the errors that you want to exclude from future Lockdown and Runtime tests.</p> <p>These settings are stored in the default exclusion list (the Exclusion file selected on the General Tab of the Options dialog box), but no changes are made to the Project file. This filter is based on the error code associated with an error, and these error codes are stored in the default exclusion list.</p>
Having these errors	<p>This list includes all the unique errors that were generated when this Test Case was executed (excluding the errors that were filtered out using the Set Filters function).</p> <p>Select an item in this list to filter the file listing by one type of error that was generated. Selecting any error will show only the corresponding errors in the list. To see all the errors, select Show All.</p>
Reset Results	Click to reset the status of all of the Test Items to Pending .

Folders View

When a Lockdown and Runtime test is run on a package executable or shortcut, a new node with the name of that executable or shortcut is listed under the **Lockdown and Runtime Test Node**. Under this executable or shortcut node, nodes for each Test Case that generated failures or has not yet been executed are listed. If any file errors were generated during this test, then the **Folders** node appears.

When the **Folders** node is selected, the Folders View opens and includes the following information:]

Table 17-19 • Lockdown and Runtime Tests View / Folders View Options

Option	Description
Test Case Status	<p>The Test Case Status area displays the state for the entire Test Case.</p> <ul style="list-style-type: none"> • If failures were generated when this Test Case was executed, QualityMonitor sets the status to Failed. • If one of the Test Cases has not yet completed, QualityMonitor sets the status to Pending. • If no failures were generated, QualityMonitor sets the status to Passed. <p>Depending on your business practices and standards, you may want to override the status of a Test Case from its current state. In this instance, you would manually select another status. In most cases, this will be setting a Test Case which QualityMonitor has marked as Failed (because one or more individual Test Items have failed) to Passed.</p>
Test Case Progress	<p>The Test Case Progress area includes the following information:</p> <ul style="list-style-type: none"> • Total test items—Number of files that were tested when the shortcut or executable was run. The number of files that were excluded is also listed. • Passed, Failed, Pending—Percentage of total test items that passed the test, failed the test, or have not yet been executed.
Comments	Enter comments to document any special considerations or facts regarding this Test Case.
Test Items List	This lists all of the folders that contained files that were executed when this Lockdown and Runtime Test was executed.
View these test items	Select one of the following to filter the listing: All , Passed , Failed , or Pending .
Set Filter	<p>Click on this button to open the Runtime Test Filters Dialog Box, which lists all errors that were generated during this test. You can then choose to select the errors that you want to exclude from future Lockdown and Runtime tests.</p> <p>These settings are stored in the default exclusion list (the Exclusion file selected on the General Tab of the Options dialog box), but no changes are made to the Project file. This filter is based on the error code associated with an error, and these error codes are stored in the default exclusion list.</p>
Having these errors	<p>This list includes all the unique errors that were generated when this Test Case was executed (excluding the errors that were filtered out using the Set Filters function).</p> <p>Select an item in this list to filter the listing by one type of error that was generated. Selecting any error will show only the corresponding errors in the list. To see all the errors, select Show All.</p>

Table 17-19 • Lockdown and Runtime Tests View / Folders View Options (cont.)

Option	Description
Reset Results	Click to reset the status of all of the Test Items to Pending .

Registry Entries View

When a Lockdown and Runtime test is run on a package executable or shortcut, a new node with the name of that executable or shortcut is listed under the **Lockdown and Runtime Test Node**. Under this executable or shortcut node, nodes for each Test Case that generated failures or has not yet been executed are listed. If any Registry Entry errors were generated during this test, then the **Registry Entries** node appears.

When the **Registry Entries** node is selected, the Registry Entries View opens and includes the following information:]

Table 17-20 • Lockdown and Runtime Tests View / Registry Entries View Options

Option	Description
Test Case Status	<p>The Test Case Status area displays the state for the entire Test Case.</p> <ul style="list-style-type: none"> • If failures were generated when this Test Case was executed, QualityMonitor sets the status to Failed. • If one of the Test Cases has not yet completed, QualityMonitor sets the status to Pending. • If no failures were generated, QualityMonitor sets the status to Passed. <p>Depending on your business practices and standards, you may want to override the status of a Test Case from its current state. In this instance, you would manually select another status. In most cases, this will be setting a Test Case which QualityMonitor has marked as Failed (because one or more individual Test Items have failed) to Passed.</p>
Test Case Progress	<p>The Test Case Progress area includes the following information:</p> <ul style="list-style-type: none"> • Total test items—Number of files that were tested when the shortcut or executable was run. The number of files that were excluded is also listed. • Passed, Failed, Pending—Percentage of total test items that passed the test, failed the test, or have not yet been executed.
Comments	Enter comments to document any special considerations or facts regarding this Test Case.
Test Items List	This lists all of the Registry Entries that were tested when this Lockdown and Runtime Test was executed.
View these test items	Select one of the following to filter the listing: All , Passed , Failed , or Pending .

Table 17-20 • Lockdown and Runtime Tests View / Registry Entries View Options (cont.)

Option	Description
Set Filter	<p>Click on this button to open the Runtime Test Filters Dialog Box, which lists all errors that were generated during this test. You can then choose to select the errors that you want to exclude from future Lockdown and Runtime tests.</p> <p>These settings are stored in the default exclusion list (the Exclusion file selected on the General Tab of the Options dialog box), but no changes are made to the Project file. This filter is based on the error code associated with an error, and these error codes are stored in the default exclusion list.</p>
Having these errors	<p>This list includes all the unique errors that were generated when this Test Case was executed (excluding the errors that were filtered out using the Set Filters function).</p> <p>Select an item in this list to filter the listing by one type of error that was generated. Selecting any error will show only the corresponding errors in the list. To see all the errors, select Show All.</p>
Reset Results	Click to reset the status of all of the Test Items to Pending .

Isolation Tests View

You can run Isolation Tests to display the location of all portable executable (PE) files (d11/ocx/exe/t1b/o1b) that are launched from a process while performing a Lockdown and Runtime test. Viewing a listing of these portable executable file names and paths makes it easier for you to ensure that the application is fully isolated.

After you perform a Lockdown and Runtime Test for an executable (.exe) or a shortcut on the [Lockdown and Runtime Tests View](#), an additional node called **Isolation Tests** is added to the tree under the executable or shortcut node.

When you select this **Isolation Tests** node, the filenames of the portable executable files and their paths are listed. By default, the status of all these items is **Pending**. To ensure that all of the executables or shortcuts in this test case are isolated, go to the **Test Case Status** area of the view, and set the status of the entire test case to either **Pending**, **Passed**, or **Failed**.



Note • The *Isolation Tests* node will be added to the Lockdown and Runtime Tests tree only if the selected executable launches at least one portable executable file.

- You can specify when you would like the **Isolation Tests** node on the **General** tab of the [Options Dialog Box](#). On the **General** tab, select an option from the **Show Isolation Tests** list.
- QualityMonitor does not support isolation testing under Windows 2000.

User-Defined Tests View

As your business practices dictate, you can add additional, custom tests to the QualityMonitor project file. This is accomplished by right-clicking on the User-Defined Tests view and selecting Add Test Case.

Test Case View

When you select a user-defined Test Case under User-Defined Tests on the View List, the Test Case View opens.

This view contains the following options:

Table 17-21 • Test Case View Options




Option	Description
Test Case Status	Specify the status of the selected Test Case by selecting one of the following options: <ul style="list-style-type: none"> • Pending—Test case has not been executed. • Passed—Test case has been executed and has passed. • Failed—Test case has been executed and has failed.
Instructions	Enter any instructions to explain how to execute this Test Case.
Select an executable	Select an executable to launch when this Test Case is run.
Comments	Enter comments to document the purpose of this Test Case or to note any important issues.

Deployment Status View

The Deployment Status View lists all of the products and features in the MSI package.




Products and features in the MSI package are listed in the Deployment Status tree, with an icon indicating its status:

Table 17-22 • Deployment Status View Status Icons

Icon	Description
	installed
	not installed
	a key file is either missing or does not match the version or size of that file recorded in the MSI file

When you select the Deployment Status node, all of the components in all of the product features are listed on the right. If you select an individual feature, only those components within that feature are listed. The following information is displayed:

Table 17-23 • Deployment Status View Options

Option	Description
Component Name	Name of all components in the MSI package or selected feature.
Component Status	Status of the listed component: either installed () , not installed () , or a key file is either missing or does not match the version or size of that file recorded in the MSI file ()
Component Location	Location of installed component.
Component ID	GUID of the component, which uniquely identifies it.

Part 5

Preparing Installations for Deployment

This part of the AdminStudio 9.5 User Guide includes the following chapters:

- [Preparing Packages for Distribution Using Distribution Wizard](#)
- [Distributing Packages Using Configuration Manager Web Console](#)
- [Analyzing Installations Prior to Deployment](#)
- [License-Enabling Packages Using FLEXwrap](#)

Preparing Packages for Distribution Using Distribution Wizard

The final step in the application migration process is preparing your packages for distribution. AdminStudio includes the Distribution Wizard to assist you in deploying your package (and any associated transforms) using one of the following distribution types:

Table 18-1 • Supported Distribution Types

Distribution Type	Topic
Administrative Install	Creating Administrative Installations for Packages
Altiris	Preparing for Altiris Distribution
FTP Location	Distributing Packages to FTP Servers
LANDesk	Preparing for LANDesk Distribution
ManageSoft	Preparing for ManageSoft Distribution
Marimba	Using Marimba Channel Publishing for Package Deployment
Network Location	Distributing Packages to Network Locations
SMS	Preparing for SMS Distribution
Configuration Manager Web Console	Distributing Packages Using the Configuration Manager Web Console
Tivoli	Preparing for Tivoli Distribution
ZENworks Configuration Management	Preparing for ZENworks Configuration Management 10 Distribution

Table 18-1 • Supported Distribution Types (cont.)

Distribution Type	Topic
ZENworks Desktop Application	Preparing for ZENworks Desktop Application Distribution
ZENworks Server Distribution	Preparing for ZENworks Server Distribution

This section also explains how to deploy a Windows installer package using an InstallShield script-based setup. See [Deploying InstallScript MSI Installations](#).

Creating Administrative Installations for Packages


In an administrative installation, the installation software is copied to a network directory using the administrative install option provided by Windows Installer.



Task: *To distribute your package (and any associated transforms) as an administrative installation:*

1. Launch the Distribution Wizard by either clicking on its icon in the Tools Gallery or by selecting a package in the Application Manager and selecting **Distribute Package** from the context menu.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **Administrative Install** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.

If you launched the Distribution Wizard from the Application Manager by selecting a package and selecting **Distribute Package** from the context menu, the name in the **Windows Installer Package (.msi)** field is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:

- **Not in the Software Repository**—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click **Browse** and select a different package.
 - **In the Software Repository**—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
 6. After specifying the package location, click **Next**. The **Administrative Install** panel opens.
 7. From the **Administrative Install** panel, specify or browse to the Network Directory to which you want to distribute the package.


8. If desired, you can use short file names during the distribution by selecting the Use short file names option.
9. Click **Next**. The **Distribution Summary** panel opens.
10. On the **Distribution Summary** panel, review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files. The **Distribution Output** panel displays progress during distribution.
11. Once the distribution finishes, click **Finish** to exit the Distribution Wizard.

Distributing Packages to FTP Servers

You can choose to distribute a package to an FTP server. If you select the **FTP Location** option on the Distribution Wizard **Distribution Type** panel, you will be required to enter the location of the FTP server, and the user name and password to connect to that server.



Task: *To distribute your package (and any associated transforms) to an FTP server:*

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **FTP Location** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **FTP Location** panel opens.
7. From the **FTP Location** panel, specify the location of the FTP server, and the user name and password to use to connect to the server. Click **Next**.
8. Review the selections you made in the **Distribution Summary** panel. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files.
9. The **Distribution Output** panel displays progress during distribution. Once the distribution finishes, click **Finish** to exit the Distribution Wizard.

Preparing for Altiris Distribution

AdminStudio 9.5 | Distribution Wizard

The Distribution Wizard supports the distribution of a setup along with any transforms and files via an Altiris Notification Server. A custom script file is required for Altiris distribution. The Distribution Wizard creates this custom script file using an XML template file that is provided: AltirisTemplate.Config.



Task: *To prepare your package for Altiris distribution:*

1. Launch the Distribution Wizard. The **Distribution Wizard Welcome** panel opens.
2. Click **Next**. The **Distribution Type** panel opens.
3. Select **Altiris** from the **Distribution Type** list and click **Next**. The **Installation Package Information** panel opens.
4. Click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the Add button in the **Additional Transforms** area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **Altiris Integration** panel opens.
7. In the **Network Directory** field, specify or browse to the network location where you want to store the installation package. The Distribution Wizard remembers the last **Network Location** that is entered and displays it the next time this panel is accessed.

The Distribution Wizard will copy the Windows Installer package along with any transforms and files to the UNC path specified. Also, the Distribution Wizard will use an XML template file (AltirisTemplate.config) to create a custom script file in this location named <packageName>.Config.



Note • You can edit AltirisTemplate.config to customize it for your organization. The file, which is installed with AdminStudio, is located in the Templates folder of the AdminStudio Shared Directory. See [Altiris XML Template](#) for more information.

8. In the Windows Installer Command Line field, enter any additional properties that you want to pass to the Windows Installer. See the [Windows Installer Property Reference](#) for more information.
9. In the **Altiris Server Location** field, enter the http: address for the location of the Altiris Server. The Distribution Wizard remembers the last Altiris Server Location that is entered and displays it the next time this panel is accessed.
10. In the **User Name** and **Password** fields, enter a User Name and Password to log onto the server entered in the **Altiris Server Location** field. The Distribution Wizard remembers the last User Name that is entered and displays it the next time this panel is accessed.
11. Click **Next**. The **Distribution Summary** panel appears, listing the selections you made in the previous panels.
12. Review the information on the **Distribution Summary** panel. If you are satisfied with them, click **Next**. The **Distribution Output** panel displays progress during distribution.
13. Once the distribution finishes, click **Finish** to exit the **Distribution Wizard**.



Note • For all distribution types, the Distribution Wizard will create a Distribution log file in the Distribution folder of the AdminStudio Shared Directory.

Preparing for LANDesk Distribution


AdminStudio 9.5 | Distribution Wizard

With LANDesk distribution, the MSI package along with all the setup files are copied to a network location.



Task:

To prepare your package for LANDesk distribution:

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **LANDesk** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **LANDesk Integration** panel opens.
7. In the Network Directory or URL field on the **LANDesk Integration** panel, specify the network location where you want to copy the MSI package and all of its setup files. The Network Directory could be a URL or a UNC path. This field will default to the last used path, and will provide a most recently used list.
8. Click **Next**. The **Distribution Summary** panel opens.
9. On the **Distribution Summary** panel, review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files. The **Distribution Output** panel displays progress during distribution.
10. Once the distribution finishes, click Close to exit the Distribution Wizard.

Preparing for ManageSoft Distribution

AdminStudio 9.5 | Distribution Wizard

To convert a Windows Installer .msi package directly into a ManageSoft-wrapped .msi package for deployment to managed devices, choose the ManageSoft distribution method on the **Distribution Type** panel of the Distribution Wizard. This will also create the necessary ManageSoft .ndp file.




Caution • To use the ManageSoft distribution method in the Distribution Wizard, your system must satisfy the requirements specified in ManageSoft 7.2 MGS.500.006 PRD as well as the AdminStudio system requirements.

Perform the following steps to prepare for ManageSoft distribution:



Task: *To prepare your package for ManageSoft distribution:*

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **ManageSoft** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **ManageSoft Distribution Settings** panel opens.
7. Select one of the following options from the Configure MSI Package Source field:
 - **Install From Cache**—Configure the ManageSoft-wrapped package to be installed from a locally managed device cache.
 - **Install From Server**—Configure the ManageSoft-wrapped package to be installed from a Server.
 - **Install Source based upon Managed Device's Configuration**—Configure the ManageSoft-wrapped package to be installed from either a Server or device cache, depending on the package's configuration.
8. Select the Retain the ManageSoft wrapped .msi package on local managed device cache option if you do not want the Distribution Wizard to delete the ManageSoft wrapped .msi package from the local cache after installation. You might want to select this option if you want to reinstall the ManageSoft wrapped .msi package on another machine. This option is enabled when you select Install Source based upon Managed Device's Configuration from the **Configure MSI Package Source** list.
9. Select the **Install MSI package with elevated privileges** option if you need elevated privileges to install the ManageSoft wrapped .msi package due to lack of user privileges in your environment. Usually this option is not selected because elevated privileges are not necessary for most installations to complete successfully.
10. Click **Next**. The **ManageSoft Package and Environmental Settings** panel opens.
11. The package Manufacturer, ManageSoft Package Name, Application Name, and Version fields are populated based upon information saved in the selected package. Make any desired changes.
12. Under Supported Environments, select those environments where this ManageSoft-wrapped package will be deployed.
13. Click **Next**. The **Distribution Summary** panel opens.
14. On the **Distribution Summary** panel, review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files. The **Distribution Output** panel displays progress during distribution.
15. Once the distribution finishes, click Close to exit the Distribution Wizard.

Using Marimba Channel Publishing for Package Deployment

AdminStudio 9.5 | Distribution Wizard

Systems Administrators use the Marimba Desktop suite of products to ensure that users in their network always have the most recent software on their local computer. Distribution Wizard makes it easy for you to efficiently distribute packages using Marimba.

In Marimba Channel Publishing, the System Administrator publishes an application (called a Channel) on a Marimba Transmitter web site. The Marimba Desktop Tuner component, installed on a user's local machine, communicates with the Transmitter web site to find out which Channels this user has subscribed to, and if any of those Channels have recently been updated.

Initially, the Marimba Desktop Tuner downloads the Channels that the user has subscribed to, installs them on the user's local machine. Then it periodically checks the Transmitter web site to see if any of this user's subscribed Channels have been updated. You can use the Distribution Wizard to create and publish a Channel (such as an installation setup) on the Transmitter web site so that the Marimba Transmitter can download a Channel to a local machine.

Here is a review of terms associated with Marimba distribution:


Table 18-2 • Marimba Distribution Terminology




Term	Description
Marimba Desktop Tuner	An application on your local computer that downloads Channels from the Transmitter, allowing you to run the Channel applications on your local computer.
Transmitter	The application that stores the original Channel before it is downloaded to your computer. It usually runs on another computer in your network.
Channel	Any application or file that the Marimba Desktop Tuner downloads to your computer. Channels are initially stored on the Transmitter and then the Marimba Desktop Tuner downloads the Channels so you can run them on your local computer.
Subscription	The process of getting a Channel on your computer. Once you subscribe to a Channel, the Marimba Desktop Tuner keeps track of it, making sure the information is up to date. The Marimba Desktop Tuner and the Channel are automatically set up to make sure the Channel stays up to date on the local computer by downloading updates at scheduled intervals.




Task: *To distribute your package (and any associated transforms) using Marimba Channel publishing:*

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.

3. On the **Distribution Type** panel, select **Marimba** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the **Browse** button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. If you want to set public MSI properties in the Windows Installer package, enter those properties in the Specify Additional MSI Properties field using the following syntax:

PROP1=value1, PROP2=value2, PROP3=value3
7. After specifying the package location and specifying any additional MSI properties, click **Next**. The **Patches** panel opens.
8. In the Patches (.msp) area:
 - If there are patches associated with the package, click the New button () and navigate to the patch you want to add.
 - Use the Up and Down arrows () to set the order in which the patches are applied to the package.
 - Use the Delete button () to delete a patch from the list.
9. After you have added all patches, click **Next**. The **Marimba Integration** panel opens.
10. On the **Marimba Integration** panel, do the following:
 - a. **Specify the directory where you want to place the contents of the new Channel**—Specify the directory where you want to place the MSI file, or click Browse to select it. This directory will be called the Channel Directory.


Note • Using Distribution Wizard, users can specify where they would like to copy an MSI package and all the other related files. The Distribution Wizard will then automatically find the Application Packager Channel URL and copy the files at the specified location. This location is called the Channel Directory. Users have a provision to specify any other Application Packager URL. The Channel Directory is also referred to as the Publish Directory. Distribution Wizard is required to be able to find the Application Packager Channel URL.
 - b. **Include all the files that are in the same directory as the Windows Installer package**—If desired, select this option. If this option is selected, all the files and folders in the same directory as the MSI package will be copied into the Channel Directory.
 - c. **Specify the destination Channel URL on the transmitter**—This is an optional field. This is the URL to the Transmitter that contains the Channel. For example, the URL to a Transmitter might be http://trans.marimba.com. The URL to a Channel on that Transmitter might be http://trans.marimba.com/MyChannel. However, if you specify a value in this field, you must also specify the Channel Copier URL field on the **Advanced Marimba Options** dialog box.
11. To specify advanced Marimba options for Channel Creation and Channel Publishing, click Advanced to access the **Advanced Marimba Options** dialog box.


- a. **Specify the Application Package URL**—The Application Package URL will be used to copy the files to the Channel directory. Accept the default application package URL or enter a new one.
 - b. **Specify the XML template file**—The XML template file sets properties for the MSI file during Marimba distribution. Accept the default XML template file that was shipped with AdminStudio, or enter a new one.
 - c. **Specify the Channel Copier URL**—If you have specified a destination Channel URL on the transmitter, then specify the URL for the Channel Copier. Distribution Wizard uses the Channel Copier URL to publish the channel to a transmitter, where it will be available to desktops.
 - d. Click **OK** to return to the **Marimba Integration** panel.
12. On the **Marimba Integration** panel, click **Next**. The **Distribution Summary** panel opens.
 13. Review the selections you made in the **Distribution Summary** panel. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files. The **Distribution Output** panel displays progress during distribution.
 14. Once the distribution finishes, click Close to exit the Distribution Wizard.

Distributing Packages to Network Locations

To distribute a package to a network directory, select the **Network Location** option on the Distribution Wizard **Distribution Type** panel.



Task: *To distribute your package (and any associated transforms) to a network location:*

1. Launch the Distribution Wizard. The Distribution Wizard **Welcome** panel opens.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **Network Location** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the Browse button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **Network Location** panel opens.
7. From the **Network Location** panel, specify or browse to the Network Directory location to which you want to distribute the package, and click **Next**. The **Distribution Summary** panel opens.
8. On the **Distribution Summary** panel, review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files. The **Distribution Output** panel displays progress during distribution.
9. Once the distribution finishes, click Finish to exit the Distribution Wizard.


Preparing for SMS Distribution

AdminStudio 9.5 | Distribution Wizard

To prepare for SMS distribution, select **SMS** on the **Distribution Type** panel of the Distribution Wizard. The Distribution Wizard will create a Package Definition File. SMS 1.2 or earlier uses a .pdf file extension, and SMS 2.0 or later uses an .sms file extension.



Task: *To prepare your package for SMS distribution:*

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **SMS** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the Browse button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **SMS Distribution** panel opens.
7. From the **SMS Distribution** panel, configure settings as necessary and click **Next**. The **Network Location** panel opens.
8. On the **Network Location** panel, specify the network location to distribution the installation package and click **Next**. The **Distribution Summary** panel appears.
9. On the **Distribution Summary** panel, review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files, to the location specified in the **SMS Distribution** panel. The **Distribution Output** panel displays progress during distribution.
10. Once the distribution finishes, click **Finish** to exit the Distribution Wizard.



Note • For detailed information on SMS template files and the variables they use, see [SMS File Templates](#).



Note • Any associated transforms are merged into the Windows Installer package before the package is distributed to SMS.

SMS File Templates

AdminStudio 9.5 | Distribution Wizard

The following templates are used when creating SMS files: `template.pdf` (for SMS 1.2) and `template.sms` (for SMS 2.0). You can find these files in the `C:\AdminStudio\Shared\Templates` folder. You can replace or modify these templates and any instances of the predefined variables (below) will be replaced.

Example

The following table displays a PDF file template in the left column, and the output of that PDF file in the right column:

Table 18-3 • Comparison of an SMS Template File and an Output File

Template	Output
<pre>[PDF] Version=1.0 [Package Definition] Product=%PRODUCT% Version=%VERSION% Comment=%MANUFACTURER% SetupVariations=Typical, Automated [Typical Setup] CommandName=Typical Installation CommandLine=msiexec /i %CMDLINE% UserInputRequired=TRUE SynchronousSystemExitRequired=TRUE SupportedPlatforms=Win 9x, Win NT (i386) [Automated Setup] CommandName=Automated Installation CommandLine=msiexec /i %CMDLINE% /q UserInputRequired=FALSE SynchronousSystemExitRequired=TRUE SupportedPlatforms=Win 9x, Win NT (i386) [Setup Package for Inventory] InventoryThisPackage=FALSE</pre>	<pre>[PDF] Version=1.0 [Package Definition] Product=Orca Version=1.50.2600.0000 Comment=Microsoft SetupVariations=Typical, Automated [Typical Setup] CommandName=Typical Installation CommandLine=msiexec /i orca.msi UserInputRequired=TRUE SynchronousSystemExitRequired=TRUE SupportedPlatforms=Win 9x, Win NT (i386) [Automated Setup] CommandName=Automated Installation CommandLine=msiexec /i orca.msi /q UserInputRequired=FALSE SynchronousSystemExitRequired=TRUE SupportedPlatforms=Win 9x, Win NT (i386) [Setup Package for Inventory] InventoryThisPackage=FALSE</pre>

Supported Variable Names

Below is a list of supported variable names and what they map to in the PDF/SMS files.

Table 18-4 • SMS Supported Variable Names

Variable Name	Value	SMS 1.2 (.PDF)	SMS 2.0 (.SMS)
%PRODUCT%	MSI ProductName	[Package Definition] Product	[Package Definition] Name

Table 18-4 • SMS Supported Variable Names (cont.)

Variable Name	Value	SMS 1.2 (.PDF)	SMS 2.0 (.SMS)
%VERSION%	MSI ProductVersion	[Package Definition] Version	[Package Definition] Version
%MANUFACTURER%	MSI Manufacturer	[Package Definition] Comment	[Package Definition] Publisher
%CMDLINE%	MSI filename + transforms	N/A	[SetupVariation] CommandLine
%MIF%	MIF filename specified in Wizard	N/A	[Package Definition] MIFFilename
%LANGUAGE%	MSI ProductLanguage converted into text	N/A	[Package Definition] Language
%PRODUCTCODE%	MSI ProductCode	N/A	[SetupVariation] UninstallKey



Note • These variable names are case sensitive.


Preparing for Tivoli Distribution

AdminStudio 9.5 | Distribution Wizard

To prepare your package for Tivoli distribution, perform the following steps.



Task: *To prepare your package for Tivoli distribution:*

1. Launch the Distribution Wizard.
2. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
3. From the **Distribution Type** panel, select **Tivoli** and click **Next**. The **Installation Package Information** panel opens.
4. On the **Installation Package Information** panel, click the Browse button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **Tivoli Integration** panel appears.

7. Specify or browse to the Network Directory where you want to store the installation package and created SPD file. Click **Next**. The **Tivoli Settings** panel opens.
8. Configure settings as necessary and click **Next**. The **SPD Parameters** panel opens.
9. Configure each feature's install state. When finished, click **Next**. The **Distribution Summary** panel opens.
10. Review the selections you made. If you are satisfied with them, click **Next** to distribute the package, including associated transforms and files, to the location specified in the Tivoli Integration panel. The **Distribution Output** panel displays progress during distribution.
11. Once the distribution finishes, click **Finish** to exit the Distribution Wizard.



Note • Any associated transforms are merged into the Windows Installer package before the package is distributed to Tivoli.

Preparing for ZENworks Configuration Management 10 Distribution

AdminStudio 9.5 | Distribution Wizard

To prepare your package for ZENworks Configuration Management 10 distribution, perform the following steps.



Task: *To prepare your package for ZENworks Configuration Management 10 distribution:*

1. Launch the Distribution Wizard for ZENworks Configuration Management by performing the following steps:
 - a. Launch the **Distribution Wizard**.
 - b. On the **Welcome** panel, click **Next**. The **Distribution Type** panel opens.
 - c. Select **ZENworks Configuration Management Distribution** and click **Next**.

The **Welcome** panel of the Distribution Wizard for ZENworks Configuration Management opens.



Note • If you do not want the **Welcome** panel to be displayed each time you open this wizard, select the **Do not show the Welcome panel again** option. If this option is selected, the **Login** panel will be the first panel opened for this wizard.

2. Click **Next** to continue. The **Login** panel opens.

- On the **Login** panel, enter the following login information for the ZENworks Configuration Management server that you are connecting to:

Property	Description
User Name / Password	Enter a valid User Name and Password for the ZENworks Configuration Management server you are connecting to.
Server URL	<p>Enter the server URL, machine name, or IP address of the ZENworks Configuration Management server using the following format:</p> <p>http://www.servername.com or http://111.22.333.44</p> <p>If you need to specify a specific port number, append the port number to the end of the URL, such as:</p> <p>http://www.servername.com:123</p> <p>If you are using SSL and you want a secure connection, change the http prefix to https. For example:</p> <p>https://www.servername.com</p>

- Click **Login**. The **Windows Installer Package Information** panel opens.
- Click **Browse** next to the **Windows Installer Package File (.msi)** field and select the Windows Installer package that will be referenced by this ZENworks Configuration Management bundle.





Important • All of the files in the selected Windows Installer file's directory and all of its subdirectories will be uploaded to the ZENworks Configuration Management Server.

After you make your selection, if there are any transform files (.mst) in the same directory, they are listed in the **Windows Installer Transform Files (.mst)** area.



Note • All of the .mst files that are in the same directory as the selected Windows Installer package are automatically listed in the **Windows Installer Transform Files (.mst)** list, even if they are not applicable to the selected package. To prevent the inclusion of non-applicable transform files, delete those transforms from the list.

- To include transforms with the Windows Installer package:
 - Click the New button () in the **Windows Installer Transform Files (.mst)** area and select a transform file. If the package requires multiple transforms, you can repeat the procedure as necessary.
 - Select a transform and click the Delete button () to delete a transform from the list.

7. If you want to customize how this package is installed, enter parameters in the **Install Parameters** field. Any actions that you enter will be performed whenever this bundle is installed.

- The root parameter, which should not be edited or deleted, is `/i packagename.msi`.
- These parameters are applied to `msiexec.exe` to perform the desired action.
- By default, the `/qn` parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.

If this operation requires user input, either remove the `/qn` parameter, or create a response transform to preconfigure all user input. For more information, see [Using Response Transforms](#).

- For additional parameters that can be added, see [Additional Install, Uninstall, and Repair Parameters](#).

8. If you want to customize how this package is uninstalled, enter parameters in the **Uninstall Parameters** field. Any actions that you enter will be performed whenever this bundle is uninstalled.

- The root parameter, which should not be edited or deleted, is `/x packagename.msi`.
- By default, the `/qn` parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.

If this operation requires user input, either remove the `/qn` parameter, or create a response transform to preconfigure all user input. For more information, see [Using Response Transforms](#).

- For additional parameters that can be added, see [Additional Install, Uninstall, and Repair Parameters](#).

9. If you want to customize how this package is repaired, enter parameters in the **Repair Parameters** field. Bundles are repaired by reinstalling missing or corrupted files.

- The root parameter, which should not be edited or deleted, is `/f packagename.msi`.
- By default, the `/qn` parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.

If this operation requires user input, either remove the `/qn` parameter, or create a response transform to preconfigure all user input. For more information, see [Using Response Transforms](#).

- You can apply any of the following additional Repair parameters after the package name:

Parameter	Description
p	Reinstalls a file if it is missing
o	Reinstalls a file if it is missing or if an older version of the file is present on the user's system
e	Reinstalls a file if it is missing or if an equivalent or older version of the file is present on the user's system

Parameter	Description
c	Reinstalls a file if it is missing or if the stored checksum of the installed file does not match the new file's value
a	Forces a reinstall of all files
u or m	Rewrite all required user registry entries
s	Overwrites any existing shortcuts
v	Runs your application from the source and re-caches the local installation package

- For additional parameters that can be added, see [Additional Install, Uninstall, and Repair Parameters](#).

10. Click **Next**. The **Bundle Creation Options** panel opens.

11. Specify whether to update an existing bundle or create a new one by selecting one of the following options:

- Create a new bundle from these Windows Installer package files**—To create a new bundle to reference this Windows Installer package, select this option.
- Update an existing bundle using these Windows Installer package files**—If you want to overwrite an existing bundle to contain this Windows Installer package, select this option, and then select an existing bundle from the tree:
 - Recommended Bundles**—This group lists the bundles that contain the same Windows Installer package as the one you selected on the [Windows Installer Package Information Panel](#).
 - All Other Bundles**—This group lists the rest of the existing bundles on the server.

12. After making your selection, click **Next** to proceed. The **Bundle Information** panel opens.

13. Enter information to specify attributes for this bundle on ZENworks Configuration Management:

Property	Properties
Bundle Name	Enter the bundle's name as you want it to appear in ZENworks Control Center (ZCC) and the ZENworks Application Launcher (on managed devices).
Version Number	Enter the bundle's version number. If you are overwriting an existing bundle, and you enter a higher version number than the bundle's original version number, the bundle will be redeployed.
Icon	Click Browse and select a shortcut icon graphic (in .ico, .gif, .jpg, .png, .bmp, or .exe format) that ZENworks Application Launcher will display on managed devices. If you do not select an icon file, the standard ZENworks bundle icon will be used.

Property	Properties
Folder	<p>From the Folder list, select the folder path that will be used by ZENworks Application Launcher when displaying the bundle on either the device's desktop or Start menu. All of the folders defined on the ZENworks server are listed. For example:</p> <ul style="list-style-type: none"> • Start Menu—If you specify Applications\Accounting as the path and choose to display the bundle on the Start menu, ZENworks Application Launcher creates an Application\Accounting folder on the root of the Start menu and adds the bundle to it. • Desktop—If you specify Applications\Accounting as the path and choose to display the bundle on the desktop, ZENworks Application Launcher creates an Applications\ Accounting folder on the desktop and adds the bundle to it. <p>You can place multiple bundles in a single folder by specifying the same folder path for each of the bundles.</p>
Description	Enter a description of the bundle. This description will be displayed in ZENworks Control Center and the ZENworks Application Launcher (on managed devices).

14. After you have entered bundle information, click **Next**. The **Summary** panel opens, displaying the options you have selected for distributing this Windows Installer package on ZENworks Configuration Management.
15. Click **Publish** to complete the distribution process. The **Publishing Process** panel opens, listing the progress messages while the bundle is being published on ZENworks Configuration Management.
 - **ZENworks error messages**—Any error messages with a numeric prefix that appear on this panel are generated by ZENworks Configuration Management. To resolve these errors, contact your ZENworks Configuration Management System Administrator.
 - **Cancelling publication**—If you want to cancel the publication of the bundle on ZENworks Configuration Management, click **Cancel**.
16. When processing is complete, the **Finish** button becomes enabled. Click **Finish** to exit this wizard.

See Also

[Distribution Wizard for ZENworks Configuration Management](#)

Preparing for ZENworks Desktop Application Distribution

AdminStudio 9.5 | Distribution Wizard


The Distribution Wizard supports the distribution of a setup along with any transforms and files via ZENworks desktop application distribution. You would select this option if you want to create an MSI Distribution Object to distribute to ZENworks desktops.



Note • If you are using ZENworks Configuration Manager Server 10, instead of using the standard Distribution Wizard, you should use the customized Distribution Wizard for ZENworks Configuration Management. See [Preparing for ZENworks Configuration Management 10 Distribution](#) for more information.



Task: To prepare your package for ZENworks desktop application distribution:

1. Launch the Distribution Wizard. The Distribution Wizard **Welcome** panel opens.
2. Click **Next**. The **Distribution Type** panel opens.
3. Select **ZENworks Desktop Application** from the **Distribution Type** list and click **Next**. The **Installation Package Information** panel opens.
4. Click the Browse button and locate the Windows Installer Package (.msi) you want to distribute.
5. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
6. After specifying the package location, click **Next**. The **ZENworks Login** panel opens.
7. Use this panel to login to the server location where you want to create the Distribution Object. Specify the following information:
 - a. In the **Login Name** field, enter a valid Login Name for the Novell server where you want to create the Distribution Object. This is a required field. For more information, see [Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel](#).
 - b. In the **Password** field, enter a valid password for the specified Login Name.
 - c. In the **Server** field, enter the machine name or IP address of the Novell server where you want to create the Distribution Object.
 - d. In the **Context** field, enter the Novell environment where you want to create the Distribution Object. Because you can choose to define the context information in the Login Name field, this field is optional. For more information, see [Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel](#).
 - e. In the **SSL Port Number** field, specify the secure port to connect to ZENworks. The default/recommended port value is 636.





Note • All of these entries except the **Password** are saved and will pre-populate these fields the next time you access this panel.

8. After you have entered the login information, click **Next**. The **ZENworks Desktop Application** panel opens.
9. Use this panel to specify the ZENworks Application Object and its properties. Specify the following information:
 - a. In the Object Name field, specify the name of the Application Object to be created in the specified Novell eDirectory® tree. You can use the name of the .msi file or any other name you choose.
 - b. In the Version field, specify the Application Object's version. For a new Application Object, leave this field empty (in which case 0 is used). For existing Application Objects, enter an integer to identify the version.
 - c. In the **Context** field, enter the Novell environment where you want to create this Application Object. For more information, see [Specifying Multiple Parameters in the Context Field of the ZENworks Server Distribution Panels](#).
 - d. In the **Administration Package Path** field, specify the path that ConsoleOne® will use to access the .msi package. ConsoleOne uses the package file to populate information in the Application Object. This field defaults to the directory specified by the .msi file. If desired, you may edit this path.



Note • This field is used only by ConsoleOne to access the .msi package. It is not used by Novell® Application Launcher for distribution of the application. For distribution, Application Launcher uses the path defined in the SOURCE_PATH.

- e. In the SOURCE_PATH area, specify a list of network directories that are the desired location(s) for the package source path(s). You can specify one or more source paths.

To add a path, click the New button () to open the Browse for Folder dialog box, and select the path. If you want to include multiple source paths, you can repeat the procedure as necessary. If you need to delete a path you have added, select the path and click the Delete button ()

- f. Click **Next**. The **Distribution Summary** panel appears, listing the selections you made in the previous panels.
10. Review the information, and if satisfied, click **Next**. The **Distribution Output** panel displays progress during distribution.
11. Once the distribution finishes, click **Finish** to exit the Distribution Wizard.



Note • For all distribution types, the Distribution Wizard will create a Distribution log file in the Distribution folder of the AdminStudio Shared directory.

Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel

The entries that you make in the **Login Name** and **Context** fields are concatenated to form the Novell distinguished name. If you enter `jsmith` in the **Login Name** field and `myorg` in the **Context** field, the login information that AdminStudio would pass to the server would be:

```
cn=jsmith, o=myorg
```

Specifying Multiple Parameters in the Login Name Field

Instead of making entries in both the **Login Name** and **Context** fields, you could enter both the login name and the context in the **Login Name** field (separated by either periods or commas). AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-5 • How To Specify Multiple Parameters in the Login Name Field

If you enter ...	AdminStudio assumes that ...
<p>two unqualified parameters in the Login field, such as:</p> <pre>jsmith, myorg</pre>	<ul style="list-style-type: none">the first entry identifies the login name (cn)the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <pre>cn=jsmith, o=myorg</pre>
<p>more than two unqualified parameters in the Login field, such as:</p> <pre>jsmith, dept01, office03, myorg</pre>	<ul style="list-style-type: none">the first entry identifies the login name (cn)the last entry identifies the organization parameter (o)all entries in between the first and the last entry are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <pre>cn=jsmith, ou=dept01, ou=office03, o=myorg</pre>

In this instance, you would leave the Context field blank.

Specifying Multiple Parameters in the Context Field

If you would like to enter multiple parameters in the Context field, you would enter the login name in the Login Name field without parameters (`jsmith`, for example), and then specify all of the context parameters in the Context field. You can enter multiple qualified parameters in any order:

```
ou=dept01, o=myorg, ou=office03
```

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-6 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
one unqualified parameter in the Context field, such as: <code>myorg</code>	The following parameter is passed to the server: <code>o=myorg</code>
two unqualified parameters in the Context field, such as: <code>dept01, myorg</code>	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) Therefore, the following parameters are passed to the server: <code>ou=dept01, o=myorg</code>
more than two unqualified parameters in the Context field, such as: <code>dept01, office03, myorg</code>	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) Therefore, the following parameters are passed to the server: <code>ou=dept01, ou=office03, o=myorg</code>



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.



Caution • The Login Name is a required field; you cannot leave the Login Name field blank and specify the login name in the Context field using the `cn` parameter.

Specifying Multiple Parameters in the Context Field of the ZENworks Server Distribution Panels

In the Context field, you can enter multiple qualified parameters in any order (separated by either periods or commas):

ou=dept01, o=myorg, ou=office03

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-7 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
one unqualified parameter in the Context field, such as: myorg	The following parameter is passed to the server: o=myorg
two unqualified parameters in the Context field, such as: dept01, myorg	<ul style="list-style-type: none">the first entry identifies an organizational unit (ou)the second entry identifies the organization parameter (o) Therefore, the following parameters are passed to the server: ou=dept01, o=myorg
more than two unqualified parameters in the Context field, such as: dept01, office03, myorg	<ul style="list-style-type: none">the last entry identifies the organization parameter (o)all previous entries are assumed to be identifying organizational units (ou) Therefore, the following parameters are passed to the server: ou=dept01, ou=office03, o=myorg



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.



Note • For all distribution types, the Distribution Wizard will create a Distribution log file in the Distribution folder of the AdminStudio Shared directory.

Preparing for ZENworks Server Distribution

AdminStudio 9.5 | Distribution Wizard

The Distribution Wizard supports the distribution of a setup along with any transforms and files via ZENworks server distribution. You would select this option if you want to create an MSI Distribution Object to distribute to ZENworks servers.




Note • If you are using ZENworks Configuration Manager Server 10, instead of using the standard Distribution Wizard, you should use the customized Distribution Wizard for ZENworks Configuration Management. See [Preparing for ZENworks Configuration Management 10 Distribution](#) for more information.



Task:

To prepare your package for ZENworks server distribution:

1. Launch the Distribution Wizard. The Distribution Wizard **Welcome** panel opens.
Click **Next**. The **Distribution Type** panel opens.
2. Select **ZENworks Server Distribution** from the **Distribution Type** list and click **Next**. The **Installation Package Information** panel opens.
3. Click the Browse button and locate the Windows Installer Package (.msi) you want to distribute.
4. If there are transforms associated with the package, click the New button () in the Additional Transforms area and navigate to the transform you want to add. Repeat as necessary.
5. After specifying the package location, click **Next**. The **ZENworks Login** panel opens.
6. Use this panel to login to the server location where you want to create the Distribution Object. Specify the following information:
 - a. In the Login Name field, enter a valid Login Name for the Novell server where you want to create the Distribution Object. This is a required field. For more information, see [Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel](#).
 - b. In the Password field, enter a valid password for the specified Login Name.
 - c. In the Server field, enter the machine name or IP address of the Novell server where you want to create the Distribution Object.
 - d. In the Context field, enter the Novell environment where you want to create the Distribution Object. Because you can choose to define the context information in the Login Name field, this field is optional. For more information, see [Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel](#).
 - e. In the SSL Port Number field, specify the secure port to connect to ZENworks. The default/recommended port value is 636.

All of these entries except the password are saved and will pre-populate these fields the next time you access this panel.
7. After you have entered the login information, click **Next**. The **ZENworks Server Distribution/Object** panel opens.
8. Use this panel to specify the ZENworks Distribution Object and its properties. Specify the following information:
 - a. In the Name field, specify the Distribution Object's name.

- ### Table 18-8 • Sample Operating System Distribution Object Paths

Operating System	Sample Distribution Object Path
Windows:	c:\packages\msi_app
Linux/Solaris:	/packages/msi_app
NetWare:	data:\packages\msi_app

- ## Specifying Multiple Parameters in the Login Name and Context Fields of the ZENworks Login Panel

cn=jsmith, o=myorg

Specifying Multiple Parameters in the Login Name Field

Instead of making entries in both the Login Name and Context fields, you could enter both the login name and the context in the Login Name field (separated by either periods or commas). AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-9 • How to Specify Multiple Parameters in the Login Name Field

If you enter ...	AdminStudio assumes that ...
<p>two unqualified parameters in the Login field, such as:</p> <p>jsmith, myorg</p>	<ul style="list-style-type: none"> the first entry identifies the login name (cn) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p>cn=jsmith, o=myorg</p>
<p>more than two unqualified parameters in the Login field, such as:</p> <p>jsmith, dept01, office03, myorg</p>	<ul style="list-style-type: none"> the first entry identifies the login name (cn) the last entry identifies the organization parameter (o) all entries in between the first and the last entry are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <p>cn=jsmith, ou=dept01, ou=office03, o=myorg</p>

In this instance, you would leave the Context field blank.

Specifying Multiple Parameters in the Context Field

If you would like to enter multiple parameters in the Context field, you would enter the login name in the Login Name field without parameters (jsmith, for example), and then specify all of the context parameters in the Context field. You can enter multiple qualified parameters in any order:

ou=dept01, o=myorg, ou=office03

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-10 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
<p>one unqualified parameter in the Context field, such as:</p> <p>myorg</p>	<p>The following parameter is passed to the server:</p> <p>o=myorg</p>

Table 18-10 • How to Specify Multiple Parameters in the Context Field (cont.)

If you enter ...	AdminStudio assumes that ...
two unqualified parameters in the Context field, such as: dept01, myorg	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> ou=dept01, o=myorg
more than two unqualified parameters in the Context field, such as: dept01, office03, myorg	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> ou=dept01, ou=office03, o=myorg



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.



Caution • The Login Name is a required field; you cannot leave the Login Name field blank and specify the login name in the Context field using the cn parameter.

Specifying Multiple Parameters in the Context Field of the ZENworks Server Distribution Panels

In the Context field, you can enter multiple qualified parameters in any order (separated by either periods or commas):

ou=dept01, o=myorg, ou=office03

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-11 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
one unqualified parameter in the Context field, such as: myorg	<p>The following parameter is passed to the server:</p> o=myorg

Table 18-11 • How to Specify Multiple Parameters in the Context Field (cont.)

If you enter ...	AdminStudio assumes that ...
<p>two unqualified parameters in the Context field, such as:</p> <p><code>dept01, myorg</code></p>	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p><code>ou=dept01, o=myorg</code></p>
<p>more than two unqualified parameters in the Context field, such as:</p> <p><code>dept01, office03, myorg</code></p>	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <p><code>ou=dept01, ou=office03, o=myorg</code></p>



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.



Note • For all distribution types, the Distribution Wizard will create a Distribution log file in the Distribution folder of the AdminStudio Shared directory.

Deploying InstallScript MSI Installations

AdminStudio 9.5 | Distribution Wizard

When deploying an InstallScript MSI installation, the file `setup.exe` needs to be deployed with the InstallScript `.msi` installation file. The `setup.exe` file is required because it launches a file (`isscriptn.msi`) that installs the InstallScript engine required to run the InstallScript code. The *n* in `installscriptn.msi` indicates the version of the InstallScript engine that was used to create the InstallScript MSI installation.

If you want to deploy an InstallScript MSI installation without using `setup.exe`, such as when using Active Directory, you need to first deploy the same version of the InstallScript engine that was used to build the InstallScript MSI installation.

Installing the InstallScript Engine

Sometimes the `isscriptn.msi` file (the file that installs the InstallScript engine) is located in the same directory as the InstallScript `.msi` file. However, in some instances, the `isscriptn.msi` file is compressed within the `setup.exe` file and cannot be accessed.

If the `isscriptn.msi` file is compressed within the `setup.exe` file, you have the following options:

- **If you know which version** of the InstallScript engine was used to create your InstallScript MSI installation, you can get a copy of the InstallScript engine from the AdminStudio installation CD. All the major releases of the InstallScript engine are available in the InstallScript_Engines folder on the AdminStudio installation CD.



Note • For more information, see the [Update to the Latest InstallShield Installation Engines Knowledge Base](#) article at

<http://consumer.installshield.com/kb.asp?id=Q108322>

- **If you do not know which version** of the InstallScript engine was used to create your InstallScript MSI installation, contact the software vendor to find out the exact version.

Deploying an InstallScript MSI Installation

To deploy an InstallScript MSI installation, configure the setup and the target system in the following manner:



Task: To deploy an InstallScript MSI installation:

1. Run the `isscriptn.msi` file to install the appropriate InstallScript engine on the target machine (where *n* is the version of the InstallScript engine that was used to create the application's InstallScript MSI installation).
2. Create a transform for the InstallScript MSI that includes the following changes:
 - a. Add the property `ISSETUPDRIVEN` to the property table via the Direct Editor and give it a value of `1`.
 - b. Add a condition to the `OnCheckSilentInstall` custom action in the `InstallExecuteSequence` via the Direct Editor that will always resolve to false or remove the custom action from the sequence.
 - c. Make any additional changes in the transform, such as populating the serial number, modifying shortcuts or feature states, depending upon your organization's needs and the features and requirements of the application.
3. Next deploy the InstallScript .msi package.

If deploying the package via Active Directory, make sure that you set the **Installation User Interface** to **Basic** and specify any transforms that you created for the InstallScript MSI package.

Reference

The Distribution Wizard provides a straightforward way to distribute a Windows Installer package (.msi)—including any associated transforms or patches—to a network location, an FTP server, an administrative location, or using virtually any distribution system.

The Distribution Wizard is comprised of two separate wizards:

- [Distribution Wizard](#)
- [Distribution Wizard for ZENworks Configuration Management](#)

Distribution Wizard

The Distribution Wizard—which is used to distribute a Windows Installer package (.msi) to a network location, an FTP server, an administrative location, or using virtually any distribution system—consists of the following panels:

- [Welcome Panel](#)
- [Distribution Type Panel](#)
 - [Administrative Install Panel](#)
 - [Altiris Integration Panel](#)
 - [FTP Location Panel](#)
 - [LANDesk Integration Panel](#)
 - [ManageSoft Distribution Settings Panel](#)
 - [Marimba Panels](#)
 - [Network Location Panel](#)
 - [SMS Distribution Panel](#)
 - [Tivoli Integration Panel](#)
 - [ZENworks Login Panel](#)
 - [ZENworks Desktop Application Panel](#)
 - [ZENworks Server Distribution/Object Panel](#)
 - [ZENworks Server Distribution/Distributor Panel](#)
- [Installation Package Information Panel](#)
- [Distribution Summary Panel](#)
- [Distribution Output Panel](#)

Welcome Panel

The Distribution Wizard provides a straightforward way to distribute a Windows Installer package (.msi)—including any associated transforms—to a network location, an FTP server, an administrative location, a Marimba channel, via a Tivoli or SMS software package definition file, to a Microsoft Configuration Manager Server, to a ZENworks Configuration Management server, to a ZENworks desktop application, to a ZENworks server, or using Altiris, LANDesk, or ManageSoft.

Distribution Type Panel

From the **Distribution Type** panel, you can select the distribution method you want to use. You can choose one of the following distribution methods:

Table 18-12 • Distribution Wizard Distribution Types

Method	Description
Administrative Install	The installation is copied to a network directory using the administrative install option provided by Windows Installer. See Creating Administrative Installations for Packages .
Altiris	Prepare for Altiris distribution. See Preparing for Altiris Distribution .
FTP Location	Distribute to an FTP server, providing both your user name and password. See Distributing Packages to FTP Servers .
LANDesk	Distribution for LANDesk involves copying the setup files to a network location. See Preparing for LANDesk Distribution .
ManageSoft	Create a new ManageSoft project for distribution. See Preparing for ManageSoft Distribution .
Marimba	A new Marimba channel is created and files are copied to it for distribution. See Using Marimba Channel Publishing for Package Deployment .
Network Location	Distribute into a network directory. See Distributing Packages to Network Locations .
SMS	The Distribution Wizard will create a Package Definition File. SMS 1.2 or earlier uses a .pdf file extension, and SMS 2.0 or later uses an .sms file extension. See Preparing for SMS Distribution .
Configuration Manager Web Console	End the Distribution Wizard and launch the Configuration Manager Web Console in a new browser window. See Distributing Packages Using Configuration Manager Web Console .
Tivoli	The Distribution Wizard copies the MSI, any MSTs, and the Tivoli SPD file to the network location specified. See Preparing for Tivoli Distribution .
ZENworks Configuration Management Distribution	Create an MSI distribution object to distribute on ZENworks Configuration Management. See Preparing for ZENworks Configuration Management 10 Distribution .
ZENworks Desktop Application	Create an MSI distribution object to distribute to ZENworks desktops. See Preparing for ZENworks Desktop Application Distribution .

Table 18-12 • Distribution Wizard Distribution Types (cont.)

Method	Description
ZENworks Server Distribution	Create an MSI distribution object to distribute to ZENworks servers. See Preparing for ZENworks Server Distribution .

Click **Next** to proceed to the associated distribution method panel.

Administrative Install Panel

This panel is displayed if you selected the Administrative Install distribution method from the **Distribution Type** panel. The installation will be copied to the network directory using the Windows Installer [administrative install](#) option.

In addition to specifying or browsing to the location where to perform the administrative installation, you can also select the Use short file names option to force the administrative installation to use the 8.3 file name convention (using the [SHORTFILENAMES](#) property).

Altiris Integration Panel

AdminStudio 9.5 | Distribution Wizard

This panel is displayed if you selected the Altiris distribution method from the Distribution Type panel. You would choose the Altiris method to create a package in the Altiris Notification Server.

Table 18-13 • Altiris Integration Panel


Field	Description
Network Directory	<p>Specify or browse to the network location where you want to store the installation package. The Distribution Wizard remembers the last Network Location that is entered and displays it the next time this panel is accessed.</p> <p>The Distribution Wizard will copy the Windows Installer package along with any transforms and files to the UNC path specified. Also, the Distribution Wizard will use an XML template file (AltirisTemplate.config) to create a custom script file in this location named <packageName>.Config.</p>  <p>Note • You can edit <i>AltirisTemplate.config</i> to customize it for your organization. The file, which is installed with AdminStudio, is located in the <i>Templates</i> folder of the AdminStudio Shared Directory. See Altiris XML Template for more information.</p>
Windows Installer Command Line	Enter any additional properties that you want to pass to the Windows Installer. See the Windows Installer Property Reference for more information.

Table 18-13 • Altiris Integration Panel

Field	Description
Altiris Server Location	Enter the http: address for the location of the Altiris Server. The Distribution Wizard remembers the last Altiris Server Location that is entered and displays it the next time this panel is accessed.
User Name	Enter a User Name to log onto the server entered in the Altiris Server Location field. The Distribution Wizard remembers the last User Name that is entered and displays it the next time this panel is accessed.
Password	Enter the password to log onto the server entered in the Altiris Server Location field.

Click **Next** to proceed to the **Installation Package Information** panel.

Altiris XML Template

AdminStudio 9.5 | Distribution Wizard

When using the Altiris distribution method, a custom script file is required. If you select the Altiris method on the [Distribution Type Panel](#), the **Distribution Wizard** uses an XML Template file (AltirisTemplate.config) to create a custom script file named <packageName>.Config. The Distribution Wizard copies this configuration file along with the Windows Installer package with any transforms and files to the Network Directory specified on the [Altiris Integration Panel](#).

You can edit AltirisTemplate.config to customize it for your organization. The file, which is installed with AdminStudio, is located in the Templates folder of the AdminStudio Shared directory.

The following variables are used in the AltirisTemplate.Config file:

Table 18-14 • AltirisTemplate.config Variables

Variable	Value
%DIST.ASVERSION%	AdminStudio version number
%DIST.COMMANDLINE%	Command line specified on the Altiris Integration Panel of the Distribution Wizard. See the Windows Installer Property Reference for more information.
%DIST.NETWORKLOCATION%	Network location of the MSI package as specified on the Altiris Integration Panel of the Distribution Wizard
%ProductCode%	ProductCode property from the MSI Property Table
%ProductName%	ProductName property from the MSI Property Table
%ProductVersion%	ProductVersion property from the MSI Property Table

Table 18-14 • AltirisTemplate.config Variables (cont.)

Variable	Value
%SUMMARYSTREAM.Id%	<p>Comments property from the MSI Summary Stream.</p> <p>Any property with the SUMMARYSTREAM prefix will be populated based on the MSI Summary Information Stream Property as specified by the Id, in the format of:</p> <pre><description>%SUMMARYSTREAM.4%</description></pre> <p>Summary Stream Ids range from 1 to 19. For a complete list of Summary Information Stream Ids, see Summary Information Stream Property Set. In the example above, "4" indicates that the value of the Author property should be inserted.</p>

Please note the following:

- Any property with the DIST prefix will be custom populated by the Distribution Wizard.
- Any other property will be populated based on the MSI Property Table.
- Typically, all variables are enclosed within '%' characters, as shown above.

FTP Location Panel

This panel is displayed when you select FTP Location as the distribution method from the **Distribution Type** panel. The installation will be upload to the FTP server specified in the **FTP Location** field. If necessary, provide a user name and password for the FTP server.

LANDesk Integration Panel

AdminStudio 9.5 | Distribution Wizard

With LANDesk distribution, the MSI package along with all the setup files are copied to a network location.

If you select LANDesk on the **Distribution Type** panel, the LANDesk Integration panel is displayed.

Specify the following options on the LANDesk Integration panel:

Table 18-15 • LANDesk Integration Panel Options

Option	Description
Network Directory or URL	Specify the network location where you want to copy the MSI package and all of its setup files. The Network Directory could be a URL or a UNC path. This field will default to the last used path, and will provide a most recently used list.

ManageSoft Distribution Settings Panel

AdminStudio 9.5 | Distribution Wizard

If you select ManageSoft on the **Distribution Type** panel, the ManageSoft Distribution Settings panel is displayed. You would choose this method to convert a Windows Installer .msi package directly into a ManageSoft-wrapped .msi package for deployment to managed devices. This will also create the necessary ManageSoft .ndp file.



Caution • To use the ManageSoft distribution method in the Distribution Wizard, your system must satisfy the requirements specified in ManageSoft 7.2 MGS.500.006 PRD as well as the AdminStudio system requirements.

Specify the following options on this panel:

Table 18-16 • ManageSoft Distribution Settings Panel Options

Option	Description
Configure MSI Package Source	<p>Choose one of the following options:</p> <ul style="list-style-type: none">• Install From Cache—Configure the ManageSoft-wrapped package to be installed from a locally managed device cache.• Install From Server—Configure the ManageSoft-wrapped package to be installed from a Server.• Install Source based upon Managed Device's Configuration—Configure the ManageSoft-wrapped package to be installed from either a Server or device cache, depending on the package's configuration.
Retain the ManageSoft wrapped .msi package on local managed device cache	<p>By default, the ManageSoft wrapped .msi package is deleted from the local cache after installation. Select this option if you do not want the Distribution Wizard to delete the ManageSoft wrapped .msi package from the local cache after installation. You might want to select this option if you want to reinstall the ManageSoft wrapped .msi package on another machine.</p> <p>This option is enabled when you select Install Source based upon Managed Device's Configuration from the Configure MSI Package Source list.</p>
Install MSI package with elevated privileges	<p>Select this option if you need elevated privileges to install the ManageSoft wrapped .msi package due to lack of user privileges in your environment. Usually this option is not selected because elevated privileges are not necessary for most installations to complete successfully.</p>

ManageSoft Package and Environmental Settings Panel

AdminStudio 9.5 | Distribution Wizard

If you select ManageSoft on the **Distribution Type** panel of the Distribution Wizard, this panel opens after you have completed the **ManageSoft Distribution Settings** panel. On this panel you are prompted to specify the package settings, and select the supported environments for this package.

Specify the following options on this panel:

Table 18-17 • ManageSoft Package and Environmental Settings Panel Options

Option	Description
Manufacturer	This will be populated by default with the Manufacturer record value in the Property table from within the original .msi package. This field can be modified.
Package	This will be populated by default with the ProductName record value in the Property table from within the original .msi package. This field can be modified.
Application Name	This will be populated by default with the ProductName record value in the Property table from within the original .msi package. This field can be modified.
Version	This field will be blank by default. This field can be modified.
Supported Environments	Select those environments where this ManageSoft-wrapped package will be deployed.



Note • All fields except Version are required to be populated in order to proceed to the next dialog. The Next button will not be enabled unless all required fields are populated.

Marimba Panels

AdminStudio 9.5 | Distribution Wizard

The following panels are displayed when you select the Marimba distribution type from the **Distribution Type** panel:

- [Patches Panel](#)
- [Marimba Integration Panel](#)
- [Advanced Marimba Options Dialog Box](#)






Note • For more information, see [Using Marimba Channel Publishing for Package Deployment](#).

Patches Panel

AdminStudio 9.5 | Distribution Wizard

This panel opens when you have selected Marimba from the **Distribution Type** panel of the Distribution Wizard.

In the Patches (.msp) area:

- If there are patches associated with the package, click the New button () and navigate to the patch you want to add.
- Use the Up and Down arrows () to set the order in which the patches are applied to the package.
- Use the Delete button () to delete a patch from the list.

Marimba Integration Panel

AdminStudio 9.5 | Distribution Wizard

This panel is displayed if you selected the Marimba distribution method from the **Distribution Type** panel. The information you specify in this panel enables you to create a directory to store contents for the new Marimba channel.

This panel contains the following options:

Table 18-18 • Marimba Integration Panel Options

Option	Description
Specify the directory where you want to place the contents of the new channel	Specify or browse to the directory where you want to place the contents of the new channel.
Channel Name	<p>This field is populated based on the .msi and .mst package name(s) selected:</p> <ul style="list-style-type: none"> • If no transforms are applied, this field is populated with the .msi file name (base name only, with .msi removed). • If a transform is applied, this field is populated with the main transform name (base name only, with .mst removed). (The first transform listed in the Additional transforms field of the Installation Package Information Panel is considered the main transform.) <p>You can edit this field before continuing. The Channel Name will be created when the Marimba Channel is created.</p>
Channel Description	This field is populated based on information in the .msi and .mst packages. You can edit this field before continuing.
Channel Version	This field is populated based on information in the .msi and .mst packages. You can edit this field before continuing. You can edit this field before continuing.
Include all the files that are in the same directory as the Windows Installer package	Select to include all files in the same directory as the installation package in the new directory. If this option is selected, all the files and folders in the same directory as the MSI package will be copied into the Channel Directory.

Table 18-18 • Marimba Integration Panel Options (cont.)

Option	Description
Specify the destination channel URL on the transmitter	This field defines the destination channel URL and subdirectory on the transmitter. This is an optional field. However, if you specify a value in this field, you must also specify the channel copier URL.
Advanced	Click to access the Advanced Marimba Options Dialog Box , where you can specify advanced Marimba options for Channel Publishing and Channel Creation.

Click **Next** to proceed.

Advanced Marimba Options Dialog Box

AdminStudio 9.5 | Distribution Wizard

On this dialog box, which is accessed by clicking the **Advanced** button on the **Marimba Integration** panel, you can specify advanced Marimba options for Channel Creation and Channel Publishing.

Specify the following options:

Table 18-19 • Advanced Marimba Options Dialog Box Options

Option	Description
Specify the Application Package URL	Accept the default application package URL or enter a new one.
Specify the XML template file	The XML template file sets properties for the MSI file during Marimba distribution. Accept the default XML template file that was shipped with AdminStudio, or enter a new one.
Specify the channel copier URL	If you have specified a destination channel URL on the transmitter on the Marimba Integration panel, then specify the URL for the channel copier in this field.

Network Location Panel

This panel is displayed when you choose the **Network Location** or **SMS** distribution methods from the **Distribution Type** panel. The installation files will be copied to the network directory you specify (or browse to) in this panel.

SMS Distribution Panel

AdminStudio 9.5 | Distribution Wizard


This panel is displayed if you selected the SMS distribution method from the **Distribution Type** panel.

For SMS distribution, you create an SMS PDF (package definition file): a text file that contains all of the information necessary to create a software distribution package for an application.



Note • For detailed information on SMS template files and the variables they use, see [SMS File Templates](#).

Table 18-20 • SMS Distribution Panel Options

Option	Description
Create Package Definition File (SMS 1.2 or earlier)	Select this option if you are using SMS 1.2 or earlier (which uses a .pdf file extension).
Create SMS File (SMS 2.0 or later)	Select this option if you are using SMS 2.0 or later (which uses a .sms file extension).
Install MIF Filename	<p>If your installation will be run in a Microsoft Systems Management Server (SMS) environment, you can create a MIF (Management Information Format) file in the Windows directory to describe your application. In this text box, insert a filename for this MIF file. This is an optional step.</p> <div></div> <p>Note • This field is enabled when the Create SMS file (SMS 2.0 or later) option is selected.</p>

Tivoli Integration Panel

AdminStudio 9.5 | Distribution Wizard

This panel is displayed if you selected the Tivoli distribution method from the **Distribution Type** panel. You can specify or browse to the network location where you want to store the installation package. All installation files—including the Tivoli SPD file—will be copied to this location.

Click **Next** to proceed.

Tivoli Settings Panel

AdminStudio 9.5 | Distribution Wizard

The **Tivoli Settings** panel appears when you select Tivoli on the **Distribution Type** panel. You use the **Tivoli Settings** panel to specify the directory from which you want to launch the product installation, enter a source image path to bundle the images in the software packages, and specify compression settings.

Specify the following options:

Table 18-21 • Tivoli Settings Panel Options

Option	Description
Target Image Path	Specify the directory from which you want to launch the product installation.
Redirect installation to a remote system	When this option is selected, product images are not included in the Tivoli package because they already exist in an accessible location with the installation.
Source Image Path	Specify the path of the directory where product images are stored. When distribution occurs, these product images are retrieved and distributed to the target systems. This path is only necessary for a bundled installation, where image files are distributed with the package.
Keep Images	Select this option to retain the installation images on the target systems after installation.
Available to All Users	Select this option if the product should be made available to all target system users. If not selected, the product will only be made available to logged-on users when the installation is run.
Compression	Specify whether you want to use deflated or stored compression for all files and directories included in the Tivoli package.

SPD Parameters Panel

AdminStudio 9.5 | Distribution Wizard

This panel allows you to set the install action for each feature in the package you are distributing. These settings will be recorded in the Software Package Definition File (SPDF).

The following install actions are possible when you click on the button to the left of the feature, although some may not be available depending on whether the feature supports the functionality:

Table 18-22 • SPD Parameters Panel Install Actions

Install Action	Description
Set feature to local	The feature will be installed when the setup is run.
Set feature and subfeatures to local	The feature and all its subfeatures will be installed when the setup is run.
Set feature to source	The feature will be run from the source media.

Table 18-22 • SPD Parameters Panel Install Actions (cont.)

Install Action	Description
Set feature and subfeatures to source	The feature and subfeatures will be run from the source media.
Set feature to advertised	The feature will be marked as advertised when the setup is run.
Set feature and subfeatures to advertised	The feature and all its subfeatures will be marked as advertised when the setup is run.
Set feature to absent	The feature will not be installed when the setup is run.



Note • For sample input and output of an SPD file, and a description of the variables used in an SPD file, see [SPD File Details](#).

SPD File Details

AdminStudio 9.5 | Distribution Wizard

A Software Package Definition File (SPDF) named template.spd will be used when creating a new SPD file (from C:\AdminStudio Shared\Templates). The .SPD file will be named after the msi file name.

Any instances of predefined variables in the file will be replaced, as described in the following table. Below the table, a [Tivoli Sample Template File](#) and a [Tivoli Sample Output File](#) are provided.



Note • In an SPD file, variables are case sensitive.

Table 18-23 • SPD File Variables

Variable	Value	SPD Location
%CREATION_TIME%	Current date/time	package\creation_time
%MODIFICATION_TIME%	Current date/time	package\last_modification_time
%ALLUSERS%	y or n	package\install_msi_product\ all_users
%REDIRECT%	y or n	package\install_msi_product\ is_image_remote
%KEEP_IMAGES%	y or n	package\install_msi_product\ keep_images
%COMPRESSION%	stored or deflated	package\install_msi_product\ compression_method

Table 18-23 • SPD File Variables (cont.)

Variable	Value	SPD Location
%IMAGE_DIR%	Image folder path	package\install_msi_product\ image_dir
%SOURCE_DIR%	Source folder path	package\install_msi_product\ source_dir
%CAPTION%	MSI file path	package\install_msi_product\ caption
%PACKAGE_FILE%	MSI file name	package\install_msi_product\ package_file
%PRODUCTCODE%	MSI ProductCode	package\install_msi_product\ product_code
%PRODUCTNAME%	MSI ProductName	package\install_msi_product\ product_name
%VERSION%	MSI ProductVersion	package\install_msi_product\ product_version
%FEATURES%	Name and install action for each MSI feature	package\install_msi_product\
%TRANSFORMS%	Sets TRANSFORMS property in MSI command line using transforms specified in Wizard.	package\install_msi_product\ properties

Tivoli Sample Template File

"TIVOLI Software Package v4.1 - SPDF"

```
package
  name = Noname
  title = "No title"
  version = 1.0
  web_view_mode = hidden
  undoable = o
  committable = o
  history_reset = n
  save_default_variables = n
  creation_time = %CREATION_TIME%
  last_modification_time = %MODIFICATION_TIME%
  move_removing_host = y
  no_check_source_host = y
  lenient_distribution = n
  default_operation = install
  server_mode = all
  operation_mode = not_transactional
```

```
post_notice = n
before_as_uid = 0
skip_non_zero = n
after_as_uid = 0
no_chk_on_rm = y
versioning_type = swd
package_type = refresh
stop_on_failure = y

install_msi_product
  caption = %CAPTION%
  image_dir = %IMAGE_DIR%
  source_dir = %SOURCE_DIR%
  is_image_remote = %REDIRECT%
  keep_images = %KEEP_IMAGES%
  compression_method = %COMPRESSION%
  package_file = %PACKAGE_FILE%
  reinstall_mode = file_equal_version,user_data,machine_data,shortcut
  log_mode = disabled,fatal_exit,error
  report_log = n
  ui_level = none
  product_code = %PRODUCTCODE%
  product_name = %PRODUCTNAME%
  product_version = %VERSION%
  all_users = %ALLUSERS%
  properties = '%TRANSFORMS%'

%FEATURES%
end
end
```

Tivoli Sample Output File

"TIVOLI Software Package v4.2 - SPDF"

```
package
  name = "MSI package"
  title = "No title"
  version = 1.0
  web_view_mode = hidden
  undoable = o
  committable = o
  history_reset = n
  save_default_variables = n
  creation_time = "2002-10-09 23:35:22"
  last_modification_time = "2002-10-09 23:46:33"
  move_removing_host = y
  no_check_source_host = y
  lenient_distribution = n
  default_operation = install
  server_mode = all
  operation_mode = not_transactional
  post_notice = n
  before_as_uid = 0
  skip_non_zero = n
  after_as_uid = 0
```



```

no_chk_on_rm = y
versioning_type = swd
package_type = refresh
stop_on_failure = y

install_msi_product
  caption = C:\miky\adminpak.msi
  image_dir = C:\miky
  source_dir = C:\miky
  is_image_remote = n
  keep_images = n
  compression_method = stored
  package_file = adminpak.msi
  reinstall_mode = file_equal_version,user_data,machine_data,shortcut
  log_mode = disabled,fatal_exit,error
  report_log = n
  ui_level = none
  product_code = {B7298620-EAC6-11D1-8F87-0060082EA63E}
  product_name = "Windows 2000 Administration Tools"
  product_version = 5.0.0.0000
  all_users = y
  properties = 'TRANSFORMS=orca.mst'

  feature
    name = Run Time Libraries
    action = default_action
  end
  feature
    name = MSI Validation Support
    action = default_action
  end
  feature
    name = Merge Module Support
    action = default_action
  end
end
end

```

ZENworks Login Panel

AdminStudio 9.5 | Distribution Wizard

This panel is displayed if you selected the ZENworks Desktop Application or ZENworks Server Distribution method from the **Distribution Type** panel.

Use this panel to login to the server location where you want to create the Distribution Object. Enter the following options:

Table 18-24 • ZENworks Login Panel Options

Option	Description
Login Name	Enter a valid Login Name for the Novell server where you want to create the Distribution Object. This is a required field. For more information, see Specifying Multiple Parameters in the Login Name and Context Fields .
Password	Enter a valid password for the specified Login Name.
Server	Enter the machine name or IP address of the Novell server where you want to create the Distribution Object.
Context	Enter the Novell environment where you want to create the Distribution Object. Because you can choose to define the context information in the Login Name field, this field is optional. For more information, see Specifying Multiple Parameters in the Login Name and Context Fields .
SSL Port Number	Specify the secure port to connect to ZENworks. The default/recommended port value is 636.



Note • All of these entries except the password are saved and will pre-populate these fields the next time you access this panel.

Specifying Multiple Parameters in the Login Name and Context Fields

The entries that you make in the Login Name and Context fields are concatenated to form the Novell distinguished name. If you enter `jsmith` in the Login Name field and `myorg` in the Context field, the login information that AdminStudio would pass to the server would be:

`cn=jsmith, o=myorg`

Specifying Multiple Parameters in the Login Name Field

Instead of making entries in both the Login Name and Context fields, you could enter both the login name and the context in the Login Name field (separated by either periods or commas). AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-25 • How to Specify Multiple Parameters in the Login Name Field

If you enter ...	AdminStudio assumes that ...
<p>two unqualified parameters in the Login field, such as:</p> <p><code>jsmith, myorg</code></p>	<ul style="list-style-type: none"> the first entry identifies the login name (cn) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p><code>cn=jsmith, o=myorg</code></p>
<p>more than two unqualified parameters in the Login field, such as:</p> <p><code>jsmith, dept01, office03, myorg</code></p>	<ul style="list-style-type: none"> the first entry identifies the login name (cn) the last entry identifies the organization parameter (o) all entries in between the first and the last entry are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <p><code>cn=jsmith, ou=dept01, ou=office03, o=myorg</code></p>

In this instance, you would leave the Context field blank.

Specifying Multiple Parameters in the Context Field

If you would like to enter multiple parameters in the Context field, you would enter the login name in the Login Name field without parameters (`jsmith`, for example), and then specify all of the context parameters in the Context field. You can enter multiple qualified parameters in any order:

`ou=dept01, o=myorg, ou=office03`

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-26 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
<p>one unqualified parameter in the Context field, such as:</p> <p><code>myorg</code></p>	<p>The following parameter is passed to the server:</p> <p><code>o=myorg</code></p>

Table 18-26 • How to Specify Multiple Parameters in the Context Field (cont.)

If you enter ...	AdminStudio assumes that ...
<p>two unqualified parameters in the Context field, such as:</p> <p>dept01, myorg</p>	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p>ou=dept01, o=myorg</p>
<p>more than two unqualified parameters in the Context field, such as:</p> <p>dept01, office03, myorg</p>	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <p>ou=dept01, ou=office03, o=myorg</p>



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.



Caution • The Login Name is a required field; you cannot leave the Login Name field blank and specify the login name in the Context field using the cn parameter.

ZENworks Desktop Application Panel

AdminStudio 9.5 | Distribution Wizard




This panel is displayed (after you log in on the **ZENworks Login** panel) if you selected the ZENworks Desktop Application distribution method from the **Distribution Type** panel. You would select this option if you want to create an MSI Distribution Object to distribute to ZENworks desktops.

This panel includes the following options:

Table 18-27 • ZENworks Desktop Application Panel Options

Option	Description
Object Name	Specify the name of the Application Object to be created in the specified Novell eDirectory® tree. You can use the name of the .msi file or any other name you choose.
Version	Specify the Application Object's version. For a new Application Object, leave this field empty (in which case 0 is used). For existing Application Objects, enter an integer to identify the version.

Table 18-27 • ZENworks Desktop Application Panel Options (cont.)

Option	Description
Context	Enter the Novell environment where you want to create this Application Object. You can enter multiple context parameters in this field. See Specifying Multiple Parameters in the Context Field .
Administration Package Path	<p>Use this field to specify the path that ConsoleOne® will use to access the .msi package. ConsoleOne uses the package file to populate information in the Application Object. This field defaults to the directory specified by the .msi file. If desired, you may edit this path.</p>  <p>Note • This field is used only by ConsoleOne to access the .msi package. It is not used by Novell® Application Launcher for distribution of the application. For distribution, Application Launcher uses the path defined in the SOURCE_PATH.</p>
SOURCE_PATH	<p>Specify a list of network directories that are the desired location(s) for the package source path(s). You can specify one or more source paths. To add a path, click the Add () button to open the Browse for Folder dialog box, and select the path. If you want to include multiple source paths, you can repeat the procedure as necessary. If you need to delete a path you have added, select the path and click the Delete button ()</p>

Specifying Multiple Parameters in the Context Field

In the Context field, you can enter multiple qualified parameters in any order (separated by either periods or commas):

`ou=dept01, o=myorg, ou=office03`

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-28 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
<p>one unqualified parameter in the Context field, such as:</p> <p><code>myorg</code></p>	<p>The following parameter is passed to the server:</p> <p><code>o=myorg</code></p>
<p>two unqualified parameters in the Context field, such as:</p> <p><code>dept01, myorg</code></p>	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p><code>ou=dept01, o=tcmyorg</code></p>

Table 18-28 • How to Specify Multiple Parameters in the Context Field (cont.)

If you enter ...	AdminStudio assumes that ...
more than two unqualified parameters in the Context field, such as: dept01, office03, myorg	<ul style="list-style-type: none">the last entry identifies the organization parameter (o)all previous entries are assumed to be identifying organizational units (ou) Therefore, the following parameters are passed to the server: ou=dept01, ou=office03, o=myorg



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.

Click **Next** to proceed.

ZENworks Server Distribution/Object Panel

AdminStudio 9.5 | Distribution Wizard

If you have selected the ZENworks Server Distribution method from the **Distribution Type** panel, this panel is displayed after you log in on the **ZENworks Login** panel. On this panel you specify the ZENworks Distribution Object and its properties.

This panel includes the following options:

Table 18-29 • ZENworks Server Distribution/Object Panel Options

Option	Description
Name	Specify the Distribution Object's name.
Context	Enter the Novell environment of this Distribution Object. You can enter multiple context parameters in this field. See Specifying Multiple Parameters in the Context Field .

Specifying Multiple Parameters in the Context Field

In the Context field, you can enter multiple qualified parameters in any order (separated by either periods or commas):

ou=dept01, o=myorg, ou=office03

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-30 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
one unqualified parameter in the Context field, such as: <code>myorg</code>	The following parameter is passed to the server: <code>o=myorg</code>
two unqualified parameters in the Context field, such as: <code>dept01, myorg</code>	<ul style="list-style-type: none"> the first entry identifies an organizational unit (ou) the second entry identifies the organization parameter (o) Therefore, the following parameters are passed to the server: <code>ou=dept01, o=myorg</code>
more than two unqualified parameters in the Context field, such as: <code>dept01, office03, myorg</code>	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) Therefore, the following parameters are passed to the server: <code>ou=dept01, ou=office03, o=myorg</code>



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.

ZENworks Server Distribution/Distributor Panel

AdminStudio 9.5 | Distribution Wizard


When you select the ZENworks Server Distribution method from the **Distribution Type** panel, this panel is displayed after the **ZENworks Server Distribution/Object** panel. On this panel, you specify the Distributor Object for the Distribution Object that you are creating.

This panel includes the following options:

Table 18-31 • ZENworks Server Distribution/Distributor Panel Options

Option	Description
Context	Enter the Novell environment where you want to create this Distributor Object. You can enter multiple context parameters in this field. See Specifying Multiple Parameters in the Context Field .

Table 18-31 • ZENworks Server Distribution/Distributor Panel Options (cont.)

Option	Description
Distributor	Specify the Distributor Object for the Distribution Object that you are creating. Click the Browse button () to select the Distributor from a list of Distributors within the current context (as specified on the ZENworks Login panel).
Path to copy to Distributor's file system	Specify the desired path on the Distributor as the location where you want to copy the setup files.
Path to reference in the Distribution object	Specify the desired path as the location where you want to populate the distribution object. For example: <ul style="list-style-type: none"> • Windows: <ul style="list-style-type: none"> • c:\packages\msi_app • Linux/Solaris: <ul style="list-style-type: none"> • /packages/msi_app • NetWare: <ul style="list-style-type: none"> • data:\packages\msi_app

Specifying Multiple Parameters in the Context Field

In the Context field, you can enter multiple qualified parameters in any order (separated by either periods or commas):

```
ou=dept01, o=myorg, ou=office03
```

However, if you enter multiple unqualified parameters in the Context field, AdminStudio uses a formula to determine how to pass these parameters to the server:

Table 18-32 • How to Specify Multiple Parameters in the Context Field

If you enter ...	AdminStudio assumes that ...
one unqualified parameter in the Context field, such as: <code>myorg</code>	The following parameter is passed to the server: <code>o=myorg</code>
two unqualified parameters in the Context field, such as: <code>dept01, myorg</code>	<ul style="list-style-type: none"> • the first entry identifies an organizational unit (ou) • the second entry identifies the organization parameter (o) <p>Therefore, the following parameters are passed to the server:</p> <p><code>ou=dept01, o=myorg</code></p>

Table 18-32 • How to Specify Multiple Parameters in the Context Field (cont.)

If you enter ...	AdminStudio assumes that ...
<p>more than two unqualified parameters in the Context field, such as:</p> <p>dept01, office03, myorg</p>	<ul style="list-style-type: none"> the last entry identifies the organization parameter (o) all previous entries are assumed to be identifying organizational units (ou) <p>Therefore, the following parameters are passed to the server:</p> <p>ou=dept01, ou=office03, o=myorg</p>



Caution • If the above formulas do not accommodate your data, always use fully qualified naming.

Click **Next** to proceed.

Installation Package Information Panel

On the **Installation Package Information** panel, you select the Windows Installer package that is ready for distribution.

Table 18-33 • Installation Package Information Panel Options





Option	Description
Windows Installer Package (.msi)	<p>Specify or browse to the Windows Installer package (.msi) that you want to distribute.</p> <p>If you launched the Distribution Wizard from the Application Manager by selecting a package and selecting Distribute Package from the context menu, the name in the Windows Installer Package (.msi) field is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:</p> <ul style="list-style-type: none"> Not in the Software Repository—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click Browse and select a different package. In the Software Repository—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed. <div style="text-align: center;">  </div> <p>Edition • The Software Repository is included in AdminStudio Enterprise Edition.</p>

Table 18-33 • Installation Package Information Panel Options (cont.)

Option	Description
Additional Transforms	<p>In the Additional Transforms area:</p> <ul style="list-style-type: none">• If there are transforms associated with the package, click the New button () and navigate to the transform you want to add.• Use the Up and Down arrows () to set the order in which the transforms are applied to the package.• Use the Delete button () to delete a transform from the list.
Specify Additional MSI Properties	<p>[Marimba Only] Enter any public MSI properties that you want to set in the Windows Installer package. Use the following syntax:</p> <p><code>PROP1=value1, PROP2=value2, PROP3=value3</code></p>

Click **Next** to proceed.

Distribution Summary Panel

The **Distribution Summary** panel displays a summary of all settings configured in the previous panels. When you click **Next**, the distribution begins and the **Distribution Output** panel is displayed.



Caution • The distribution will overwrite the contents of the distribution folder.

Distribution Output Panel

The **Distribution Output** panel displays a progress bar and status messages during distribution. When distribution completes, click **Finish** to exit the Wizard.



Note • For all distribution types, the Distribution Wizard will create a Distribution log file in the Distribution folder of the AdminStudio Shared directory.

Distribution Wizard for ZENworks Configuration Management

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

You can use the Distribution Wizard for ZENworks Configuration Management to distribute a Windows Installer package (.msi)—including any associated transforms—to ZENworks Configuration Management.

The Distribution Wizard for ZENworks Configuration Management consists of the following panels:

- [Welcome Panel](#)
- [Login Panel](#)
- [Windows Installer Package Information Panel](#)
- [Bundle Creation Options Panel](#)
- [Bundle Information Panel](#)
- [Summary Panel](#)
- [Publishing Process Panel](#)

Welcome Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

You can use the Distribution Wizard for ZENworks Configuration Management to prepare a Windows Installer package (.msi)—including any associated transforms—for distribution on ZENworks Configuration Management.

If you do not want this panel to be displayed each time you open this wizard, select the **Do not show the Welcome panel again** option. If this option is selected, the **Login** panel will be the first panel opened for this wizard.

Click **Next** to continue.

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Login Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

On the **Login** panel, enter the login information for the ZENworks Configuration Management server that you want to distribute packages on, and click **Login** to proceed with the distribution process.

Enter the following information:

Table 18-34 • ZENworks Configuration Management Server Login Information

Property	Description
User Name	Enter a valid User Name for the ZENworks Configuration Management server where you want to distribute packages
Password	Enter a valid Password for the ZENworks Configuration Management server where you want to distribute packages

Table 18-34 • ZENworks Configuration Management Server Login Information (cont.)

Property	Description
Server URL	<p>Enter the server URL, machine name, or IP address of the ZENworks Configuration Management server using the following format:</p> <p>http://www.servername.com or http://111.22.333.44</p> <p>If you need to specify a specific port number, append the port number to the end of the URL, such as:</p> <p>http://www.servername.com:123</p> <p>If you are using SSL and you want a secure connection, change the http prefix to https. For example:</p> <p>https://www.servername.com</p>

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Windows Installer Package Information Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

On the **Windows Installer Package Information** panel, enter the information that will be referenced by this ZENworks server bundle, and click **Next** to continue.

Enter the following information:

Table 18-35 • Windows Installer Package Information Panel Properties

Property	Description
Windows Installer Package file (.msi)	Click Browse and select the Windows Installer (.msi) package that you want to distribute.

Table 18-35 • Windows Installer Package Information Panel Properties (cont.)






Property	Description
Windows Installer Transform Files (.mst)	<p>All of the .mst files that are in the same directory as the selected Windows Installer package are automatically listed in this list.</p> <p>To include transforms with the Windows Installer package, click the New button () and select a transform. If the package requires multiple transforms, you can repeat the procedure as necessary.</p> <p>Use the Delete button () to delete the selected transform from the list.</p>  <p>Note • All of the .mst files that are in the same directory as the selected Windows Installer package are automatically listed in this list even if they are not applicable to the selected package. To prevent the inclusion of non-applicable transform files, delete those transforms from the list.</p>
Install Parameters	<p>You can customize how this package is installed by entering parameters in this field. These parameters are applied to <code>msiexec.exe</code> to perform the desired action. Any actions that you enter here will be performed whenever the bundle is installed.</p> <p>The root parameter, which should not be edited or deleted, is:</p> <pre>/i packagename.msi</pre> <p>By default, the <code>/qn</code> parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.</p>  <p>Caution • If this operation requires user input, either remove the <code>/qn</code> parameter, or create a response transform to preconfigure all user input. For more information, see Using Response Transforms.</p>  <p>Note • For additional parameters that can be added, see Additional Install, Uninstall, and Repair Parameters.</p>

Table 18-35 • Windows Installer Package Information Panel Properties (cont.)





Property	Description
Uninstall Parameters	<p>Enter an action that will be performed whenever the bundle is uninstalled.</p> <p>The root parameter, which should not be edited or deleted, is:</p> <pre>/x packagename.msi</pre> <p>By default, the /qn parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.</p> <div></div> <div><p>Caution • If this operation requires user input, either remove the /qn parameter, or create a response transform to preconfigure all user input. For more information, see Using Response Transforms.</p><div></div><p>Note • For additional parameters that can be added, see Additional Install, Uninstall, and Repair Parameters.</p></div>

Table 18-35 • Windows Installer Package Information Panel Properties (cont.)

Property	Description
Repair Parameters	<p>Enter an action that will be performed whenever the user chooses to repair the bundle by repairing or reinstalling missing or corrupted files.</p> <p>The root parameter, which should not be edited or deleted, is:</p> <pre>/f packagename.msi</pre> <p>By default, the <code>/qn</code> parameter is added to indicate that you want to perform this operation silently (with no user interface). This is the common operating behavior for installing software with ZENworks. Running an operation silently implies that it does not require any user input.</p>  <p>Caution • If this operation requires user input, either remove the <code>/qn</code> parameter, or create a response transform to preconfigure all user input. For more information, see Using Response Transforms.</p> <p>You can also apply any of the following additional parameters after the package name:</p> <ul style="list-style-type: none"> • p – Reinstalls a file if it is missing • o – Reinstalls a file if it is missing or if an older version of the file is present on the user's system • e – Reinstalls a file if it is missing or if an equivalent or older version of the file is present on the user's system • c – Reinstalls a file if it is missing or if the stored checksum of the installed file does not match the new file's value • a – Forces a reinstall of all files • u or m – Rewrite all required user registry entries • s – Overwrites any existing shortcuts • v – Runs your application from the source and re-caches the local installation package  <p>Note • For additional parameters that can be added, see Additional Install, Uninstall, and Repair Parameters.</p>

Additional Install, Uninstall, and Repair Parameters

The following additional parameters can be entered in the Parameters fields.

Table 18-36 • Additional Parameters

Parameter	Description
<pre>/j [u m] packagename.msi /j [u m] packagename.msi /t <transform list> /j [u m] packagename.msi /g /j <language ID></pre>	<p>Building with the /j <package> option advertises the components of your application on the end user's computer</p> <ul style="list-style-type: none"> • u – Advertises components only to the current user • m – Advertises components to all users of the computer • g – Specifies language ID • t – Applies a transform to your advertised product <p>Transforms allow the synchronization of applications across different languages. For example, if you upgrade the English version of your product, you could apply a transform to automatically upgrade the French version of your product.</p>
<pre>/L [i w e a r u c m p v +] <log file></pre>	<p>Building with the /L option specifies the path to the log file. These flags indicate which information to record in the log file:</p> <ul style="list-style-type: none"> • i – Logs status messages • w – Logs non-fatal warning messages • e – Logs any error messages • a – Logs the commencement of action sequences • r – Logs action-specific records • u – Logs user requests • c – Logs initial user interface parameters • m – Logs out-of-memory messages • p – Logs terminal settings • v – Logs the verbose output setting • + – Appends to an existing file • * – Is a wildcard character that allows you to log all information (excluding the verbose output setting)

Table 18-36 • Additional Parameters (cont.)

Parameter	Description
/q [n b r f]	<p>The /q option is used to set the user interface level in conjunction with the following flags:</p> <ul style="list-style-type: none"> • q or qn – Creates no user interface • qb – Creates a basic user interface <p>The user interface settings below display a modal dialog box at the end of installation:</p> <ul style="list-style-type: none"> • qr – Displays a reduced user interface • qf – Displays a full user interface • qn+ – Displays no user interface • qb+ – Displays a basic user interface
/y <filename>	This command calls the DllRegisterServer entry-point function of the DLL or OCX file specified in <filename>.
/z <filename>	This command calls the DllUnregisterServer entry-point function of the DLL or OCX file specified in <filename>.
TRANSFORMS	<p>Use the TRANSFORMS command-line parameter to specify any transforms that you would like applied to your base package. Your transform command-line call might look something like this:</p> <pre>msiexec /i "C:\Directory\ProductName.msi" TRANSFORMS="New Transform 1.mst"</pre> <p>You can separate multiple transforms with a semicolon. Because of this, it is recommended that you do not use semicolons in the name of your transform, as the Windows Installer service will interpret those incorrectly.</p>
Properties	<p>All public properties can be set or modified from the command line. Public properties are distinguished from private properties by the fact that they are in all capital letters. For example, COMPANYNAME is a public property.</p> <p>To set a property from the command line, use the following syntax: PROPERTY=VALUE.</p> <p>If you wanted to change the value of COMPANYNAME, you would enter:</p> <pre>msiexec /i "C:\Directory\ProductName.msi" COMPANYNAME="YourCompany"</pre>

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Bundle Creation Options Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

On the **Bundle Creation Options** panel, specify whether you want to create a new bundle or overwrite an existing bundle. After making your selection, click **Next** to proceed.

You have the following options:

Table 18-37 • Bundle Creation Options

Option	Description
Create a new bundle from these Windows Installer package files	To create a new bundle to reference this Windows Installer package, select this option.
Update an existing bundle using these Windows Installer package files	<p>If you want to overwrite an existing bundle to reference this Windows Installer package, select this option, and then select an existing bundle in the tree:</p> <ul style="list-style-type: none">• Recommended Bundles—This group lists the bundles that contain the same Windows Installer package as the one you selected on the Windows Installer Package Information Panel.• All Other Bundles—This group lists the rest of the existing bundles on the server.

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Bundle Information Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

On the **Bundle Information** panel, enter information to specify attributes for this bundle on ZENworks Configuration Management, and click **Next** to continue.

Enter the following properties:

Table 18-38 • Bundle Information Panel Properties

Property	Properties
Bundle Name	Enter the bundle's name as you want it to appear in ZENworks® Control Center (ZCC) and the ZENworks Application Launcher (on managed devices).
Version Number	Enter the bundle's version number. If you are overwriting an existing bundle, and you enter a higher version number than the bundle's original version number, the bundle will be redeployed.

Table 18-38 • Bundle Information Panel Properties (cont.)

Property	Properties
Icon	Click Browse and select a shortcut icon graphic (in .ico, .gif, or .jpg format) that ZENworks Application Launcher will display on managed devices. If you do not select an icon file, the standard ZENworks bundle icon will be used.
Folder	<p>From the Folder list, select the folder path that will be used by ZENworks Application Launcher when displaying the bundle on either the device's desktop or Start menu. All of the folders defined on the ZENworks server are listed. For example:</p> <ul style="list-style-type: none"> • Start Menu—If you specify Applications\Accounting as the path and choose to display the bundle on the Start menu, ZENworks Application Launcher creates an Application\Accounting folder on the root of the Start menu and adds the bundle to it. • Desktop—If you specify Applications\Accounting as the path and choose to display the bundle on the desktop, ZENworks Application Launcher creates an Applications\Accounting folder on the desktop and adds the bundle to it. <p>You can place multiple bundles in a single folder by specifying the same folder path for each of the bundles.</p>
Description	Enter a description of the bundle. This description will be displayed in ZENworks® Control Center and the ZENworks Application Launcher (on managed devices).

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Summary Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

The **Summary** panel displays the options you have selected for distributing this Windows Installer package on ZENworks Configuration Management.

Click **Publish** to complete the distribution process or **Back** to change the listed options.

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Publishing Process Panel

AdminStudio 9.5 | Distribution Wizard for ZENworks Configuration Management

The **Publishing Process** panel lists the progress messages while the bundle is being published on ZENworks Configuration Management.

- **ZENworks error messages**—Any error messages with a numeric prefix that appear on this panel are generated by ZENworks Configuration Management. To resolve these errors, contact your ZENworks Configuration Management System Administrator.
- **Cancelling publication**—If you want to cancel the publication of the bundle on ZENworks Configuration Management, click **Cancel**.
- **Exiting the wizard**—When processing is complete, the **Finish** button becomes enabled. Click **Finish** to exit this wizard.

See Also

[Preparing for ZENworks Configuration Management 10 Distribution](#)

Distributing Packages Using Configuration Manager Web Console

You can use Microsoft System Center Configuration Manager Web Console, a Web-based application, to configure package distribution options and distribute setup packages without going to Configuration Manager Server. By using the Configuration Manager Web Console, you can:

- Create and distribute packages to Configuration Manager Server from Configuration Manager Web Console.
- View the package deployment status from Configuration Manager Web Console after it is targeted to distribute using Configuration Manager.
- Choose the target audience for deployment by browsing through the Configuration Manager collections.
- Configure program options for packages and target multiple distribution points for packages.
- Refresh source files on the distribution points if source files are changed.
- Create access accounts for setup packages to secure source files.
- Delete the packages.



Caution • *Configuration Manager Web Console features cannot be installed on 64 bit machines because Microsoft .NET Framework 2.0 does not support 64 bit machines.*

Information on how to use the Configuration Manager Web Console is presented in the following sections:

Table 19-1 • Configuration Manager Web Console Help Library

Section	Description
About Microsoft System Center Configuration Manager and Configuration Manager Web Console	Explains the concepts behind Microsoft System Center Configuration Manager and Configuration Manager Web Console and how to get started using Configuration Manager Web Console.

Table 19-1 • Configuration Manager Web Console Help Library

Section	Description
Distributing Packages Using the Configuration Manager Web Console	Explains how to configure package distribution options and distribute setup packages using Configuration Manager Web Console. <ul style="list-style-type: none">• Configuring Distribution Settings• Distributing a New Package• Viewing the Status of a Distributed Package• Modifying the Distribution Settings of a Package• Deleting Packages
Configuration Manager Web Console Reference	Every view and dialog box available in Configuration Manager Web Console is discussed in this section.

For information on how to use the Configuration Manager Web Console, click on one the following topics:

- [About Microsoft System Center Configuration Manager and Configuration Manager Web Console](#)
- [Distributing Packages Using the Configuration Manager Web Console](#)
- [Configuration Manager Web Console Reference](#)

About Microsoft System Center Configuration Manager and Configuration Manager Web Console

AdminStudio is directly integrated with Microsoft System Center Configuration Manager via the Configuration Manager Web Console, which enables enterprises using the Configuration Manager distribution system to cut down software rollout times by greatly simplifying their distribution efforts enterprise-wide.

You can use Configuration Manager Web Console to prepare, publish, and distribute applications through your Configuration Manager system without ever touching the Configuration Manager Server console, helping you improve your application management efforts while simplifying your overall Configuration Manager management requirements. You can use Configuration Manager Web Console to perform all of your Configuration Manager administrative tasks, including publishing and advertising applications and retiring or deleting applications from circulation. Configuration Manager Web Console also allows the use of existing Configuration Manager collections to distribute your applications, which simplifies targeting.

Using the centrally managed Configuration Manager Web Console, you can better monitor any application's Configuration Manager deployment success using a single interface, simplifying the process and saving you time.

The following topics explain the concepts behind Microsoft Configuration Manager and Configuration Manager Web Console and how to get started using Configuration Manager Web Console.

- [Distributing Software Through Configuration Manager Server](#)
- [Configuration Manager Web Console Overview](#)
- [Getting Started](#)

Distributing Software Through Configuration Manager Server

Microsoft Configuration Manager is used by Systems Administrators to maintain and upgrade software for organizations with distributed networks. Configuration Manager enables you to upgrade and configure each computer from a central location or from multiple locations, and schedule software programs for distribution to specific computers.

Software distribution with Configuration Manager includes the following tasks:

Table 19-2 • Software Distribution with Microsoft Configuration Manager Server

#	Step	Description
1	Install Client Agent Software on Client Computers	Install the Advertised Programs Client Agent on the Client computers to enable them to receive and run programs from Configuration Manager Server (advertised programs).
2	Prepare Site System Roles	<p>Prepare the following Site System Roles:</p> <ul style="list-style-type: none">• Client Access Points—A site system role that provides a communication point between the site server and Legacy Client computers (computers using a Windows operating system that is pre-Windows 2000). Computers contact CAPs to install and update Configuration Manager Legacy Client software. After Configuration Manager Legacy Client software has been installed on computers in a site, they contact a CAP for updated information from the Configuration Manager site server.• Management Points—Management points are the primary point of contact between Advanced Configuration Manager Clients and the Configuration Manager site server.• Distribution Points—Distribution points are site system roles that store software package files so clients can access them during the software distribution process. Clients contact distribution points to obtain source files when they run programs that are advertised to them through a Client Access point or a Management Point.

Table 19-2 • Software Distribution with Microsoft Configuration Manager Server

#	Step	Description
3	Prepare Collections	A collection is a customized management view of the resources in your organization. Collections serve as targets for performing software distribution on multiple resources at one time. An example of a collection is All Windows NT Workstation 4.0 Systems. The clients that need to receive the program must be members of a collection (referred to as the target collection).
4	Create Software Distribution Packages	Software Distribution Packages are applications packages that you are going to distribute via Configuration Manager Server. Package properties include the software name and version, the location of the package source folder, and group permissions for the distribution folder.
5	Specify Distribution Points	Select the Distribution Points that will receive the software.
6	Create Programs	Create a program for the package that is going to be distributed. Program properties specify the command line that will run on the clients, and the types of clients on which the program can run. A package can have multiple programs defined. For example, you can define a Full Installation, a Custom Installation, and a Minimum Installation for a single package.
7	Create Advertisements	Advertise the program that you want the clients to run. Advertising the program makes a program available to a specified target collection. The advertisement contains the name of the program, the name of the target collection, and the scheduling configuration (such as when to run the program or when will the program expire).

Configuration Manager Web Console Overview

Configuration Manager Web Console enables you to coordinate and manage the packaging, preparation, and distribution of applications across your enterprise via your Configuration Manager system without ever touching the Configuration Manager Server console. The following diagram explains the role that the Configuration Manager Web Console plays in Configuration Manager software distribution:

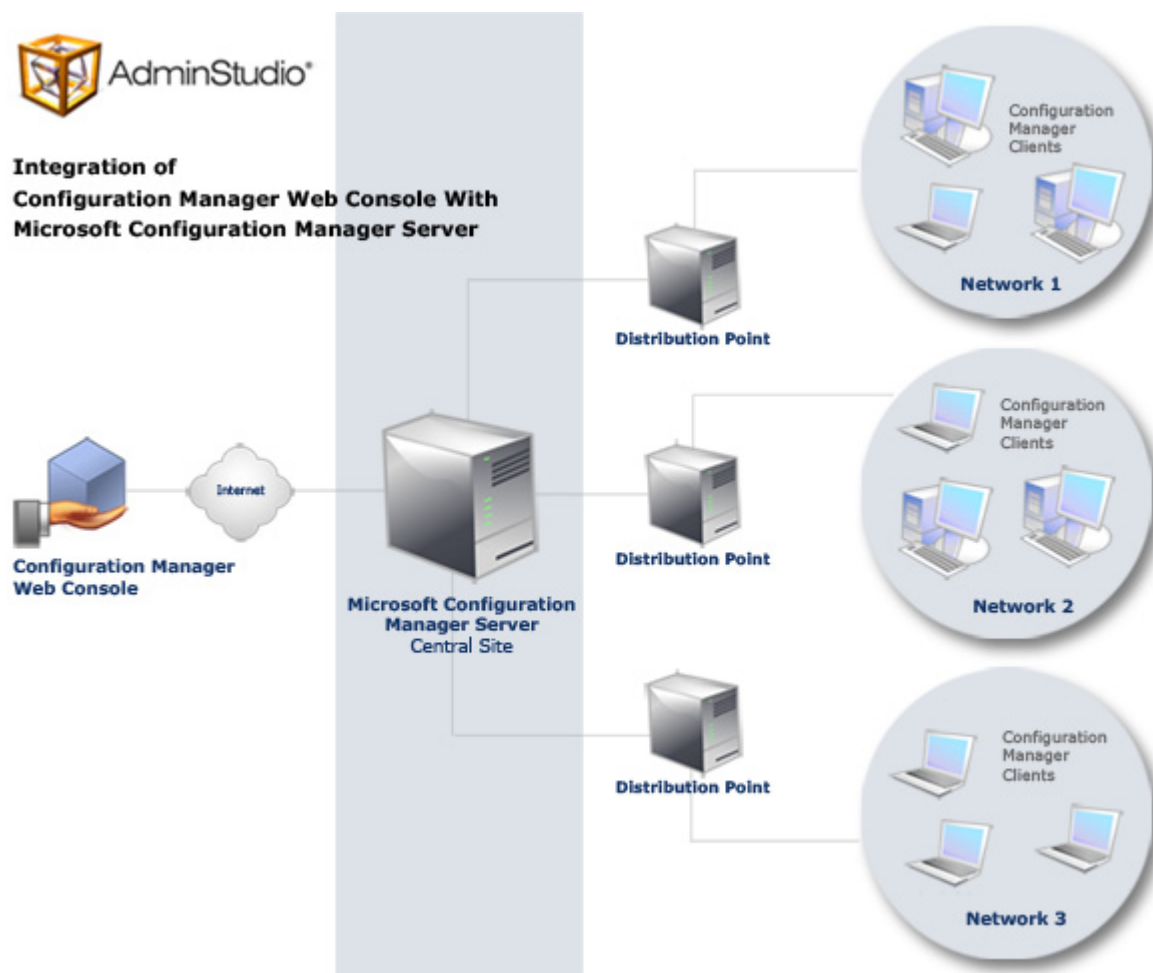


Figure 19-1: Configuration Manager Web Console and Configuration Manager Server Integration

Setting Configuration Options

In the Package Configuration area of Configuration Manager Web Console, you specify distribution configuration fields that the Configuration Manager Server utilizes in the distribution of the package. You can manually enter package and distribution configuration information and settings, or you can choose to attach a Package Definition File (.pdf) to a Windows Installer (.msi) package to pre-populate these fields. The following configuration information is specified:

- **Access Accounts**—Specify which administrators are allowed to distribute this particular package.
- **Distribution Points**—Identify which servers may act as distribution points for this particular package.

- **Programs**—Specify programs (that will run on the clients) to execute the distribution of the application, and the types of clients on which the program can run (directories, operating system versions, etc.)
- **Advertisements**—Identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory.

See [Setting Package Configuration Options](#) for detailed information on setting these options.

Package Administration

The Package Administration area of Configuration Manager Web Console allows you to perform the following tasks:

- [Viewing the Status of a Distributed Package](#)
- [Modifying the Distribution Settings of a Package](#)
- [Deleting Packages](#)

Getting Started

To start using the Configuration Manager Web Console to distribute packages to Configuration Manager Server, you need to do the following:

- **Learn About Configuration Manager Web Console**—Review the [About Microsoft System Center Configuration Manager and Configuration Manager Web Console](#) section to learn how Configuration Manager Web Console is integrated with Configuration Manager Server.
- **Configure Distribution Settings**—Perform the steps in [Configuring Distribution Settings](#) to set the default Configuration Manager Server values for any new package you target for distribution.

You would then be ready to perform the steps in [Distributing a New Package](#).

Distributing Packages Using the Configuration Manager Web Console

When you launch the Configuration Manager Web Console, the [Microsoft System Center Configuration Manager Web Console Home Page](#) opens listing three options: **Distribute a New Package**, **Package Administration**, and **Distribution Settings**. The following procedures explain how to use these options to configure package distribution options and distribute packages without using the Configuration Manager Server Console:

- [Configuring Distribution Settings](#)
- [Distributing a New Package](#)
- [Viewing the Status of a Distributed Package](#)
- [Modifying the Distribution Settings of a Package](#)
- [Deleting Packages](#)

Configuring Distribution Settings

You configure the Configuration Manager Server connection settings on the [Distribution Settings Page](#).



Note • The values entered on the **Distribution Settings** page will be used as the default values for any new package you target for distribution. However, you can change these values while configuring the package.



Task: **To configure distribution settings:**

1. Launch Configuration Manager Web Console. The **Configuration Manager Web Console** Home Page opens.
2. Click **Distribution Settings**. The **Distribution Settings** page opens.
3. In the **Site Server Name** field, enter the machine name on which Configuration Manager Server is installed. (This could be a local or remote machine.) When you make a selection in this field, the **Site Code** field is populated with a list of sites available on the Configuration Manager Server.
4. In the **Site Code** field, select the Configuration Manager site you want to use for software distribution.
5. In the **User Name** field, enter the user name of a user who has access to the Configuration Manager Server.
6. Click **Update** to save your entries
7. Click the **Home** link to return to the **Configuration Manager Web Console** Home Page.

Distributing a New Package

You can use Configuration Manager Web Console to create packages, configure package distribution options, and distribute setup packages without going to Configuration Manager Server. These package configuration settings are saved with the application in the Application Catalog, enabling you to easily redistribute the package using the same settings.

Distributing a package using Configuration Manager Web Console consists of the following tasks:

Table 19-3 • Configuration Manager Web Console Package Distribution Steps

#	Step	Description
1	Selecting a Package for Distribution	Select the package in your Application Catalog that you want to distribute, and choose whether you want to attach a Package Definition File (.pdf). When you use a Package Definition File, appropriate fields on the Configuration Manager Package Configuration Page are populated with the information in the package definition file you selected.

Table 19-3 • Configuration Manager Web Console Package Distribution Steps

#	Step	Description
2	Setting Package Configuration Options	<p>In the Package Configuration area of Configuration Manager Web Console, you specify the following distribution configuration fields that the Configuration Manager Server utilizes in the distribution of the package.</p> <ul style="list-style-type: none"> • Setting Package Settings Options—General information about the package, along with data source, data access, distribution settings, and reporting information. • Setting Access Accounts Options—Specify which administrators are allowed to distribute this particular package. • Setting Distribution Points Options—Specify which servers may act as distribution points for this particular package. A Distribution Point is a site system role that stores software package files so clients can access them during the software distribution process. • Setting Programs Options—Specify programs (that will run on clients) to execute the distribution of the application, and the types of clients on which the program can run (defined using directories, operating system versions, etc.). • Setting Advertisements Options—Identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory.
3	Saving the Package on the Configuration Manager Server	<p>Save the package on the Configuration Manager Server. You can then view the status of the package by following the instructions in Viewing the Status of a Distributed Package.</p>

Selecting a Package for Distribution

To select a package for distribution from Configuration Manager Web Console using Configuration Manager Distribution, perform the following steps.



Task:

To distribute a new package:

1. Launch Configuration Manager Web Console. The **Configuration Manager Web Console** Home Page opens.
2. Click Distribute a New Package. The **Distribute Package** page opens, listing all of the Windows Installer (.msi) packages in the open Application Catalog.
3. Click the radio button next to the Application Name of the package that you want to distribute. Only one package can be selected at a time.
4. Click **Distribute**. The **PDF Selection** page opens.

5. In the **Package Configuration** area of Configuration Manager Web Console, you specify distribution configuration fields that the Configuration Manager Server utilizes in the distribution of the package. You can manually enter package and distribution configuration information and settings, or you can choose to attach a Package Definition File (.pdf) to a Windows Installer (.msi) package to pre-populate these fields.

A Package Definition File (.pdf) is a text file that contains predefined programs and property settings for a package and is used to automate package creation. When you use a package definition file, appropriate fields on the [Configuration Manager Package Configuration Page](#) are populated with the information in the package definition file you selected. A package definition file typically has an .sms file name extension (.pdf in previous versions of Configuration Manager).

If you want to distribute this application using an existing Configuration Manager Package Definition File (.pdf or .sms) (either one on the Configuration Manager Server or one on the machine where Configuration Manager Web Console is installed), select a Package Definition File by performing the following steps:

- a. Select the **Distribute using Package Definition** option if you want to distribute this application using an existing Configuration Manager Package Definition File (.pdf or .sms). When you select this check box, the **Use Existing Package Definition** option is automatically selected, and the existing Package Definitions available on the Configuration Manager Server are listed.
- b. Select the **Use Existing Package Definition** option to distribute this application using any Package Definitions available on the Configuration Manager Server. When the **Distribute using Package Definition** check box is selected, this option is automatically selected, and the existing Package Definitions available on the Configuration Manager Server are listed. Select a Package Definition File from the list.
- c. Select the **Use Package Definition from the following file** option if you want to distribute this application using a .pdf or .sms file located on the machine where Configuration Manager Console is installed. When you select this option, all of the .pdf or .sms files in the following directory are listed:

```
<Configuration Manager Console Installation Directory>\SMS Web Console  
Application\Distribution Providers\SMS\PDFs
```

6. Select a Package Definition File from the list.
7. Click **Next**. The [Configuration Manager Package Configuration Page](#) appears, listing the information in the .pdf or .sms file (if one was selected).
8. Proceed with the steps in [Setting Package Configuration Options](#) to either enter or modify the package configuration options.

Setting Package Configuration Options

On the [Configuration Manager Package Configuration Page](#) of Configuration Manager Web Console, you specify distribution configuration fields that the Configuration Manager Server utilizes in the distribution of the package. You can manually enter package and distribution configuration information and settings, or you can choose to attach a Package Definition File (.pdf) to a Windows Installer package to pre-populate these fields.



Task: *To set package configuration options:*

1. Perform the steps described in [Selecting a Package for Distribution](#).
2. Specify configuration options on each of the following Views of the [Configuration Manager Package Configuration Page](#):
 - a. **Package Settings View**—This View opens when a package name is selected in the tree. Specify general information about the package, along with data source, data access, distribution settings, and reporting information. See [Setting Package Settings Options](#).
 - b. **Access Accounts View**—This View opens when Access Accounts is selected in the tree. Specify which administrators are allowed to distribute this particular package. See [Setting Access Accounts Options](#).
 - c. **Distribution Points View**—This View opens when Distribution Points is selected in the tree. Identify which servers may act as distribution points for this particular package. See [Setting Distribution Points Options](#).
 - d. **Programs View**—This View opens when Programs is selected in the tree. Specify programs (that will run on the clients) to execute the distribution of the application, and the types of clients on which the program can run (directories, operating system versions, etc.). See [Setting Programs Options](#).
 - e. **Advertisements View**—This View opens when Advertisements is selected in the tree. Identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory. See [Setting Advertisements Options](#).



Caution • Remember to click **Update** on each view to save your changes.

3. When you have finished specifying configuration options, click **Next**. The **Package Summary** page opens, providing a summary of all of the options specified for the package.
4. Proceed with the steps in [Saving the Package on the Configuration Manager Server](#).

Setting Package Settings Options

On the [Package Settings View](#), you specify general information about the package, along with data source, data access, distribution settings, and reporting information.



Task: *To set Package Settings configuration options:*

1. Perform the steps described in [Selecting a Package for Distribution](#).
2. On the [Configuration Manager Package Configuration Page](#), select the package name in the tree. The [Package Settings View](#) opens, presenting information on five tabs: **General**, **Data Source**, **Data Access**, **Distribution Settings**, and **Reporting**.

3. Enter the following information on the **General** tab.


Option	Description
Icon	To change the displayed icon, select a different icon from the list. This list is populated with the files available from: <Configuration Manager Web Console Installation Directory>\SMS Web Console Application\Distribution Providers\SMS\Icons
Name	The name of the package, up to 50 characters. This field is required.
Version	The version number of the software package, up to 32 characters.
Publisher	The name of the software publisher, up to 32 characters.
Language	The language version of the software, up to 32 characters.
Comment	Optional text about the package, such as a description. You can use up to 127 characters.

4. Click the **Data Source** tab and enter the following information:

Option	Description
This Package contains source files	This package has source files that are required to run its programs when they are advertised to clients. By default, this check box is cleared. When this check box is cleared, Configuration Manager does not use distribution points for the package.
Source Directory	The network path to the package source files.
Use a compressed copy of the source directory	<p>Configuration Manager will create a compressed version of the source files on the site server. When you specify additional or refresh distribution points for the package, the compressed version will be decompressed and copied to the distribution points, rather than being copied directly from the original source directory.</p> <p>Use this option if the source files might be removed from the specified path (for example, if the source files are on a compact disc). A compressed version of the source directory is always created and used when the package is sent to distribution points in child sites.</p> <p>This field is available only when the This Package contains source files check box is selected.</p>

Chapter 19: Distributing Packages Using Configuration Manager Web Console


Distributing Packages Using the Configuration Manager Web Console

Option	Description
Always obtain files from source directory	<p>Configuration Manager will obtain package source files (the files needed to run the package programs, such as .exe and .dll files) from the path specified in the Source directory box. Use this option if the source files are likely to remain at the specified path, such as a share on a server.</p> <p>This field is available only when the This package contains source files check box is selected. By default, this option is selected.</p>  <p>Note • This option is unavailable if the package is a compressed version replicated from a parent site.</p>

5. Click the **Data Access** tab and enter the following information:

Property	Description
Access distribution folder through common Configuration Manager package share	Users will access the package through the common Configuration Manager package share (SMSPKGx\$, where x is the drive letter) on distribution points. By default, this option is selected.
Share Distribution folder	The package will be created on distribution points with an administrator-specified share (and path name) that is entered in the Share name text box.
Disconnect users from distribution points	<p>Notifies users before disconnecting them from distribution points when Configuration Manager updates package data. By default, this check box is cleared.</p> <ul style="list-style-type: none">• Number of retries before disconnecting users—The number of times that Configuration Manager tries to update the package source files before starting to disconnect users who are connected to the distribution point. The default setting is 2; you can specify 0 to 99, inclusive.• User grace period—The number of minutes Configuration Manager should wait after notifying users before disconnecting them from distribution points. The default setting is 5 minutes; you can specify between 0 and 59 minutes, inclusive.

6. Click the **Distribution Settings** tab and enter the following information:

Option	Description
Sending priority	<p>The priority of this package when sent to distribution points in child sites. Packages can be sent with High, Medium, or Low priority. The default setting is Medium priority.</p> <ul style="list-style-type: none"> • If a package has High priority, it will be sent before packages with Medium or Low priority. • If a package has Low priority, it will be sent after packages with higher priority settings. • If packages have identical priorities, they will be sent in the order in which they were created in the Configuration Manager Administrator console. <p>Sending priority for a package is carried over in packages sent from a parent site to a child site.</p>
Preferred sender	<p>The sender to be used to send this package to distribution points in other sites. You can specify any sender configured for the site, or No Preference (which uses any available sender). The default setting is No Preference.</p>  <p>Note • To use Courier Sender to distribute packages, you must select it as the preferred sender.</p>

7. Click the **Reporting** tab and enter the following information:

Property	Description
Use package properties for status MIF matching	Configuration Manager will use the properties in the General tab for status MIF file matching. By default, this option is selected.
Use these fields for status MIF matching	<p>Configuration Manager will use the following values specified for status MIF file matching:</p> <ul style="list-style-type: none"> • MIF file name—The name of the MIF file that contains the package status, up to 50 characters. • Name—The name of the package, up to 50 characters. • Version—The version number of the package, up to 32 characters. • Publisher—The software publisher of the package, up to 32 characters.

8. When you have finished entering the Package Settings options, proceed with entering the rest of the Package Configuration options:
- [Setting Access Accounts Options](#)
 - [Setting Distribution Points Options](#)

- [Setting Programs Options](#)
 - [Setting Advertisements Options](#)
9. If you are finished entering Configuration Manager Package Configuration options, click **Next** to proceed to the **Package Summary** page and perform the steps listed in [Saving the Package on the Configuration Manager Server](#).

Setting Access Accounts Options

On the [Access Accounts View](#), you specify which administrators are allowed to distribute this particular package.



Task: *To set Access Accounts configuration options:*

1. Perform the steps described in [Selecting a Package for Distribution](#).
2. On the [Configuration Manager Package Configuration Page](#), select **Access Accounts** in the tree. The [Access Accounts View](#) opens.
3. Use this view to create new users and groups. If you select a user's icon under Access Accounts in the tree, you can review (but not change) that user's permissions and can delete the user.

Option	Description
Domain User Listing	<p>List of all defined users, displaying the following:</p> <ul style="list-style-type: none"> • Name—User name using the syntax of Domain\User. • Type—Identifies whether this is a user or a group, and identifies whether they have Windows or Generic access. • Permissions—Lists the user or group's permission.
Create New Access Account	<p>To create a new access account, select whether you want to create a Windows User Access Account or a Generic Access Account, and perform the following steps:</p> <p>To create a Windows User Access Account:</p> <ol style="list-style-type: none"> 1. Enter a User Name using the syntax of Domain\User. 2. Under Account Type, select User or Group. 3. Select this user's permission level from the Permissions list. 4. Click Create. <p>To create a Generic Access Account:</p> <ol style="list-style-type: none"> 1. Under Account Type, select Users, Guests or Administrators. 2. Select this generic account's permission level from the Permissions list. 3. Click Create.

4. When you have finished entering the Access Accounts options, proceed with entering the rest of the Package Configuration options:
 - [Setting Package Settings Options](#)
 - [Setting Access Accounts Options](#)
 - [Setting Distribution Points Options](#)
 - [Setting Programs Options](#)
 - [Setting Advertisements Options](#)
5. If you are finished entering Configuration Manager Package Configuration options, click **Next** to proceed to the **Package Summary** page and perform the steps listed in [Saving the Package on the Configuration Manager Server](#).

Setting Distribution Points Options

On the **Distribution Points View**, specify which servers may act as distribution points for this particular package. A Distribution Point is a site system role that stores software package files so clients can access them during the software distribution process. See [Distributing Software Through Configuration Manager Server](#) for more information.



Task: *To set Distribution Points options:*

1. Perform the steps described in [Selecting a Package for Distribution](#).
2. On the [Configuration Manager Package Configuration Page](#), select Distribution Points in the tree. The [Distribution Points View](#) opens, listing targeted distribution points and displaying the following information:
 - **Name**—The name of the distribution point.
 - **Site**—The name of the site where the distribution point is located (for example, NYC - New York City).
 - **Type**—The type of distribution point (for example, Windows NT Server or NetWare Bindery).
3. To target a new distribution point(s), select a distribution point(s) from the **Select New Distribution Point** list and click **OK**. The new distribution point will now be listed under **Distribution Points** in the tree.
4. When you have finished entering the Distribution Points options, proceed with entering the rest of the Package Configuration options:
 - [Setting Package Settings Options](#)
 - [Setting Access Accounts Options](#)
 - [Setting Programs Options](#)
 - [Setting Advertisements Options](#)
5. If you are finished entering Configuration Manager Package Configuration options, click **Next** to proceed to the **Package Summary** page and perform the steps listed in [Saving the Package on the Configuration Manager Server](#).

Setting Programs Options

On the [Programs View](#), you specify programs (that will run on the clients) to execute the distribution of the application, and the types of clients on which the program can run (directories, operating system versions, etc.). The fields shown on the **Programs View** depends upon what is selected in the tree:


- To view a listing of existing programs and to create a new one, use the Main Programs View, which is displayed when you select **Programs** in the tree.
- To edit an existing program, select the program icon under **Programs** in the tree to access the detailed tab views.





Task: *To set Program configuration options:*


1. Perform the steps described in [Selecting a Package for Distribution](#).
2. On the [Configuration Manager Package Configuration Page](#), select **Programs** in the tree. The [Programs View](#) opens.
3. To create a new program, select **Programs** in the tree, enter a name in the **Name** text box, and click **Create**. The new program will now be listed under **Programs** in the tree. To edit the program settings on the detailed tab view, select the program in the tree.
4. To edit an existing program, select the program icon under **Programs** in the tree to access the detailed tab views (**General**, **Requirements**, **Environment**, and **Advanced**) where you can edit the program settings.
5. Click the **General** tab, and enter the following information:


Option	Description
Icon	To change the displayed icon, select a different icon from the list. This list is populated with the files available from: <Configuration Manager Web Console Installation Directory>\SMS Web Console Application\Distribution Providers\SMS\Icons
Name	The name of the program, up to 50 characters. This field is required and must be unique within a package. This name is used when defining advertisements.
Comment	Optional text about the program, such as a description. You can use up to 127 characters.

Option	Description
Command Line	<p>The command line for the program, up to 511 characters. This field is required. File name extensions are strongly recommended.</p> <p>Click Browse to navigate to the file. By default, the package source directory appears, if it exists and is accessible. When the program is run on a client, the command line file name will first be searched for within the package.</p> <ul style="list-style-type: none"> • If the file does not exist there, Configuration Manager searches the local Windows directory. • If Configuration Manager cannot find the file, the program fails. • If the file name does not have a file name extension specified, Configuration Manager tries .pif, .com, .exe, and .bat. • If the file name has an extension but is not an executable, Configuration Manager tries to apply a local association. For example, if the command line is readme.gif, Configuration Manager will start the application specified on the client for opening .gif files. For example: <pre>setup.exe /a</pre> <pre>command.com /c copy Jan98.dat c:\sales\Jan98.dat</pre>  <p>Note • On 16-bit clients, batch files <i>MUST</i> specify the .bat file name extension, or the command line will fail.</p>
Start in	<p>The executable directory for the program, up to 127 characters. This directory can be an absolute path on the client, or a path relative to the distribution point folder that contains the package. This field is optional. For example:</p> <pre>c:\office97</pre> <pre>i386</pre>
Run	<p>The program mode. By default, programs run in Normal mode. You can also run a program in Minimized, Maximized, or Hidden mode.</p>

Option	Description
After Running	<p>The action that occurs after this program is completed successfully.</p> <ul style="list-style-type: none"> • No action required—No restart or logoff is required. This is the default value. • Program restarts computer—The program requires a restart and performs the restart automatically. • Configuration Manager restarts computer—Configuration Manager restarts the computer. • Configuration Manager logs user off—Configuration Manager logs the current user off the computer after the program is completed. Use this option if the program requires that the user log off and then log on again before it can run.  <p>Note • On 16-bit clients, only No action required is supported.</p>  <p>Caution • Both the Configuration Manager restarts computer and the Configuration Manager logs user off actions are done forcefully after providing the user with a grace period. Applications that are running on clients when this occurs will not have an opportunity to save their states or data.</p>


6. Click the **Requirements** tab and enter the following information:

Option	Description
Estimated disk space	A whole number greater than or equal to zero that represents the amount of disk space the software will require when installed on the client. Also, specify the units for the value.
Maximum allowed run time	A whole number greater than zero that represents the time (in minutes) required to run the program on the client.
Notify user if the program runs 15 minutes longer than estimated	<p>Configuration Manager will monitor the run time of the program and warn the user if the program is taking longer to run than the estimated time. By default, this check box is cleared.</p>  <p>Note • This feature is not supported on 16-bit clients.</p>
This program can run on any platform	The program can run on any platform. If this option is selected, Configuration Manager does not check the platform type when advertising the program to clients. By default, this option is selected.





Option	Description
This program can run only on specified client platforms	<p>The processors, operating systems, and service packs on which this program will run. When this option is selected, at least one platform must also be selected. By default, no platforms are selected. Configuration Manager uses this information when evaluating which clients in a collection are to receive the advertised program.</p>  <p>Note • If you advertise this program to clients in Configuration Manager 1.2 sites, Configuration Manager performs a less in-depth level of platform checking than it does for Configuration Manager 2.0 clients. Platform checking for Configuration Manager 1.2 clients is limited to Windows 3.1, Windows 95, x86 Windows NT, Alpha Windows NT, and Macintosh.</p>
Additional Requirements	<p>Other information for clients, up to 127 characters. This field is optional. For example, you can notify users that they should shut down all other applications before running this program.</p>


7. Click the **Environment** tab and enter the following information:

Option	Description
Program can run	<p>Specify whether the program requires that a user be logged on to the client computer to run:</p> <ul style="list-style-type: none"> • Only when a user is logged on—Prevents the program from running if no user is logged on to the computer. This is the default setting. Use this option for clients that do not run a Windows NT operating system. • Only when no user is logged on—Prevents the program from running until the user logs off the computer. This option is valid only for Windows NT clients. <p>This option forces the program to run under the local Administrator account on the client. If a user logs on while the installation is running, the program continues to run.</p> <ul style="list-style-type: none"> • Whether or not a user is logged on—Enables the program to run with or without user interaction. This option is valid only for Windows NT clients. <p>This option forces the program to run under the local Administrator account on the client.</p>
User input required	<p>Requires that the user interact with the program when it runs. This check box is selected when Program can run is set to Only when a user is logged on. If the program does not require user input, clear this check box.</p>

Option	Description
Run mode	<p>Select one of the following run modes:</p> <ul style="list-style-type: none"> • Run with user's rights • Run with administrative rights <p>To specify that the program requires administrative privileges on the computer in order to run, select Run with administrative rights. This option is available when Program can run is set to Only when a user is logged on. By default, this check box is cleared. This option is valid only for Windows NT clients.</p> <p>The Run with administrative rights option forces the program to run under the local Administrator account on the client.</p>
Use software installation account	<p>Run the program under an administrator-specified account (the Configuration Manager Windows NT Client Software Installation Account) on Windows NT clients. This check box is available only when Program can run is set to Only when no user is logged on or Whether or not a user is logged on. By default, this check box is cleared.</p> <p>This option should be used only when the program requires access to network resources other than the package files on the distribution point.</p>
Drive mode	<p>Select the drive mode used for network connections to the distribution points.</p> <ul style="list-style-type: none"> • Runs with UNC name—Enables the program to run with a universal naming convention (UNC) name. This is the default setting. • Requires drive letter—Indicates that the program requires a drive letter to fully qualify its location, but Configuration Manager can use any available drive letter on the client. • Requires specific drive letter—Indicates that the program requires the drive letter you specify (A to Z) to fully qualify its location (for example, H). If the specified drive letter is already used on a client, the program will not run.
Reconnect to Distribution point at logon	<p>The computer will reconnect to the drive when the user logs on. By default, this check box is cleared.</p>  <hr/> <p>Note • This feature is not supported on 16-bit clients.</p>

8. Click the **Advanced** tab and enter the following information:

Option	Description
Run another program first	<p>This program requires another program (in another package that is targeted for distribution using Configuration Manager Web Console or in the current package) to run before it. By default, this check box is cleared.</p>  <p>Note • This feature is not supported on 16-bit clients.</p>  <p>Note • The program you specify to run first does not need to be advertised separately.</p>
Package	The package that contains the program that must run before this program.
Program	<p>The program that must run first.</p>  <p>Note • If the depended-upon program fails on a client, the depending program will not run. Also, if the program that runs first has been run successfully, Configuration Manager will not automatically re-run it, even if the depending program is on a recurring schedule.</p>
When this program is assigned to a computer	<p>When Program can run (in the Environment tab) is set to Only when a user is logged on, you can specify how the program is assigned to users.</p> <ul style="list-style-type: none"> • Run once for first user who logs on—The program should run only once on the computer. This is the default setting. For example, a virus checker only needs to run once on the client. • Run once for every user who logs on—The program should run once for each new user who logs on.  <p>Note • On 16-bit clients, advertised programs will run only for the first user who logs on.</p>

Option	Description
Remove software when it is no longer advertised	<p>Indicates that Configuration Manager will remove this program from client computers when it is no longer advertised to those clients. This option is mainly useful for shared network applications and is available only to programs that register with Add/Remove Programs in Control Panel.</p> <p>To use this option, enter the name of the Uninstall Registry Key the program creates in the following registry key:</p> <pre>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Uninstall</pre> <p>After all advertisements for the program are revoked from the client, Configuration Manager looks up the program's uninstall command line in the UninstallString or QuietUninstallString property in this registry key. By default, this check box is cleared.</p> <p>An advertisement is considered revoked from a client when any of the following are true:</p> <ul style="list-style-type: none"> • The client is no longer a member of the collection specified in the advertisement. • The advertisement is deleted. • The advertisement expires. • Uninstall registry key • The registry key that contains the uninstall command for this program. You can use up to 127 characters.  <p>Note • This feature is not supported on 16-bit clients.</p>
Disable this program on computers where it is advertised	<p>Temporarily disables all advertisements that contain this program. The program will be removed from the list available for users to run and will not be run via assignment until it is re-enabled. By default, this check box is cleared.</p>

- When you have finished entering the **Programs** options, proceed with entering the rest of the Package Configuration options:
 - Setting Package Settings Options
 - Setting Access Accounts Options
 - Setting Distribution Points Options
 - Setting Advertisements Options
- If you are finished entering Configuration Manager Package Configuration options, click **Next** to proceed to the **Package Summary** page and perform the steps listed in [Saving the Package on the Configuration Manager Server](#).


Setting Advertisements Options

On the [Advertisements View](#), you identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory.




Task: To set Advertisements configuration options:


1. Perform the steps described in [Selecting a Package for Distribution](#).
2. On the [Configuration Manager Package Configuration Page](#), select **Advertisements** in the tree. The [Advertisements View](#) opens, displaying a list of existing advertisements:
 - **Name**—Name of the advertisement
 - **Program**—The program to run on the client.
 - **Collection**—The target collection of computers, users, or user groups that will receive the program.
 - **Available After**—Date after which the advertisement will be available
 - **Expires After**—Expiration date of advertisement
 - **Advertisement ID**—Number uniquely identifying the advertisement
 - **Status**—Advertisement's status
3. To create a new advertisement, enter a name in the **Name** text box and click **Create**. The new advertisement will now be listed under Advertisements in the tree.
4. To edit an existing advertisement, select it under **Advertisements** in the tree, and edit it on the detailed tab view (**General** and **Schedule**).
5. On the **General** tab, enter the following information:

Option	Description
Name	The name of the advertisement, up to 50 characters. This field is required.
Comment	Optional text about the advertisement, such as a description. You can use up to 127 characters.
Program	The program (within the current package) to advertise to clients. This field is required.
Collection	<p>Select the existing collection that will receive the advertisement. Click Browse to open the Select a Configuration Manager Collection Dialog Box, where you can select the Configuration Manager collection that you want to use in this Advertisement definition. This field is required.</p>  <p>Note • Configuration Manager will not advertise a program to 16-bit clients based on user groups or user accounts, but this is supported for 32-bit clients.</p>

Option	Description
Include members of subcollections	Specifies that the program also be advertised to clients and users in the specified collection's subcollections. By default, this check box is selected.

6. On the **Schedule** tab, enter the following information

Option	Description
Advertisement start time	<p>The date and time when the program is advertised and available to run on clients. By default, the current date and time is specified.</p> <p>Greenwich Mean Time—Sets the time independent of time zones. That is, if you set the time to 1:00 P.M. and select this option, the program will be advertised on all clients simultaneously—1:00 P.M. for clients in the Greenwich mean time zone, 10:00 A.M. for clients in the Eastern time zone, and 7:00 A.M. for clients in the Pacific time zone, and so on. This check box is cleared by default, which enables you to specify a relative time for advertising the program. Configuration Manager will treat the specified time as a local time. In this case, if the time is set to 1:00 P.M., the program will be advertised on clients at 1:00 P.M. in each time zone.</p>
Mandatory assignments (Assign immediately after this event)	<p>If you want to ensure that the program runs on all clients (users cannot choose not to run the program), select:</p> <ul style="list-style-type: none"> • As soon as possible • Logon • Logoff  <p>Note • Recurring assignments are not supported on 16-bit clients. Programs advertised to 16-bit clients run only once on those clients.</p>
Assignments are not mandatory over slow links	Suspends assignments for clients on a slow link. By default, this check box is selected. (This option is available only if at least one assignment is defined.)

Option	Description
Allow users to run the program independently of assignments	<p>Enables users to run the program independent of when it is assigned. By default, this check box is cleared. (This option is available only if at least one assignment is defined.)</p> <p>Typically, only non-assigned programs are visible to users in the Configuration Manager client interface. Use this option to display an assigned program in the client interface. For example, to require users to upgrade their Microsoft Office software, you specify an assignment in the advertisement properties. But if you want to enable users to run the installation program before it is scheduled to run automatically, select this option so that the assigned program appears in Advertised Programs Wizard in the Control Panel on client computers.</p>
Advertisement will expire	<p>Advertises the program to clients only for a limited time. By default, this option is cleared. This feature is not supported on 16-bit clients; the program appears until it runs on the client.</p>  <p>Note • When a program expires, it no longer appears in Advertised Programs Monitor or Advertised Programs Wizard in Control Panel on client computers. To specify the cutoff date and time, click the arrows next to the date and time. The default values are six months from the current date and time.</p> <p>Greenwich Mean Time—Set the time independent of time zones. That is, if you set the time to 1:00 P.M. and select this option, the program will expire at all computers simultaneously—1:00 P.M. for clients in the Greenwich mean time zone, 10:00 A.M. for clients in the Eastern time zone, and 7:00 A.M. for clients in the Pacific time zone, and so on. This check box is cleared by default, which enables you to specify a relative time. Configuration Manager will treat the specified time as a local time. In this case, if the time is set to 1:00 P.M., the program will expire at 1:00 P.M. on clients in each time zone.</p>
Priority	The advertisement's priority when sent to child sites: High, Medium, or Low. The default priority is Medium.

7. When you have finished entering the **Advertisements** options, proceed with entering the rest of the Package Configuration options:
 - [Setting Package Settings Options](#)
 - [Setting Access Accounts Options](#)
 - [Setting Distribution Points Options](#)
 - [Setting Programs Options](#)
8. If you are finished entering Configuration Manager Package Configuration options, click **Next** to proceed to the **Package Summary** page and perform the steps listed in [Saving the Package on the Configuration Manager Server](#).

Saving the Package on the Configuration Manager Server

To distribute an application from Configuration Manager Web Console using Configuration Manager Distribution, perform the following steps.



Task: *To save the package to the Configuration Manager Server:*

1. Perform the steps described in [Setting Package Configuration Options](#).
2. When the **Package Summary** page appears, review the summary of the all the options specified for the package.
3. To save this package on the Configuration Manager Server, click **Commit Changes to Configuration Manager**. A message will appear informing you that the package information was successfully saved to Configuration Manager.

You could now view the status of the package by following the instructions in [Viewing the Status of a Distributed Package](#).

Viewing the Status of a Distributed Package

Click **Package Administration** on the **Configuration Manager Web Console Home** page to see the status of a distributed package, delete a package, or modify a package's distribution settings.



Task: *To view the status of a distributed package:*

1. Launch Configuration Manager Web Console. The **Configuration Manager Web Console Home Page** opens.
2. Click **Package Administration**. The **Package Administration** page appears, listing all packages which are targeted for distribution.
3. Select a package and click **View Status**. The **Connect to Configuration Manager** page appears.
4. Enter a valid **User Name** and **Password** for the Configuration Manager Server you want to connect to, and click **Next**. Configuration Manager Web Console retrieves the status of that package from the Configuration Manager Server, and displays that information on the [Package Status Page](#).

Modifying the Distribution Settings of a Package

Click **Package Administration** on the **Configuration Manager Web Console Home** Page to see the status of a distributed package, delete a package, or modify a package's distribution settings.



Task: *To modify the distribution settings of a package:*

1. Launch Configuration Manager Web Console. The **Configuration Manager Web Console** Home Page opens.
2. Click Package Administration. The **Package Administration** page appears, listing all packages which are targeted for distribution.
3. Select a package and click **Modify Settings**. The **Connect to Configuration Manager** page appears.

If this package has not been distributed, you can change the **Site Server** and **Site Code** fields on the **Connect to Configuration Manager** page. However, if the package has been distributed, you cannot change these fields.
4. Enter a valid **User Name** and **Password** for the Configuration Manager Server you want to connect to, and click **Next**. The **General** tab of the **Package Settings View** of the **Configuration Manager Package Configuration** page opens.
5. Edit the package's distribution settings as described in [Setting Package Configuration Options](#).

Deleting Packages

From the Configuration Manager Web Console, you can delete a package from the Application Catalog, from the Configuration Manager Server, or both.



Caution • Deleting a distributed package from Configuration Manager will also delete its programs and any advertisements of the programs. If the package has source files, they will be removed from distribution points. If there are access accounts for this package, they will be deleted. Also, Configuration Manager Administrators' security rights to the package will be deleted.



Caution • When you delete a package from the Application Catalog, all package configuration information is lost, even if you do not delete it from Configuration Manager. If you delete a package from the Application Catalog but not from Configuration Manager, the package would still be available on the Configuration Manager Server, but all connection between the Application Catalog and Configuration Manager is lost. Should you want to distribute that package again in the future, you would have to re-import the application into the Application Catalog, and then re-enter package configuration information on the [Configuration Manager Package Configuration Page](#).



Task: *To delete a package:*

1. Launch Configuration Manager Web Console. The **Configuration Manager Web Console** Home Page opens.
2. Click **Package Administration**. The **Package Administration** page appears, listing all packages in your Application Catalog. In the **Status** column, each package's status is either **Distributed** or **Not Distributed**.
3. Select a package and click **Delete Distribution**. The **Connect to Configuration Manager** page appears.
4. Enter a valid **User Name** and **Password** for the Configuration Manager Server you want to connect to, and click **Next**. The **Confirm Delete** page appears prompting you to confirm that you want to delete this package from the Application Catalog. If you are deleting a Distributed package, you are also prompted to confirm that you want to delete this package from Configuration Manager.
5. Select **Yes** or **No** to indicate your deletion preferences:

Package Status	Delete from Catalog?	Delete from Configuration Manager?
Not Distributed	Choose Yes to confirm the deletion from the Application Catalog or choose No to decline the deletion. If you choose Yes , the package will be deleted from the Application Catalog, along with all of the package configuration information that you entered on the Configuration Manager Package Configuration page.	N/A
Distributed	Choose Yes to confirm the deletion from the Application Catalog, or choose No to decline the deletion. If you delete a package from Configuration Manager but do not delete it from the Application Catalog, all of the package configuration information that you entered on the Configuration Manager Package Configuration page is saved, making it easy to distribute this package again in the future.	Choose Yes to confirm the deletion from Configuration Manager, or choose No to decline the deletion. Unless you are certain that you will never want to distribute this package again, it is recommended that if you delete a package from Configuration Manager, you do not delete it from the Application Catalog.

6. Click **OK** to confirm your selections.

Configuration Manager Web Console Reference

This section describes each of pages that you might encounter when using the Configuration Manager Web Console. The following pages and views are documented:

- [Configuration Manager Package Configuration Page](#)
 - [Package Settings View](#)
 - [Access Accounts View](#)
 - [Distribution Points View](#)
 - [Programs View](#)
 - [Advertisements View](#)
- [Connect to Configuration Manager Page](#)
- [Confirm Delete Page](#)
- [Distribute New Package Page](#)
- [Distribute Package Page](#)
- [Microsoft System Center Configuration Manager Web Console Home Page](#)
- [Package Administration Page](#)
- [Package Status Page](#)
- [Package Summary Page](#)
- [PDF Selection Page](#)
- [Select a Configuration Manager Collection Dialog Box](#)
- [Distribution Settings Page](#)

Microsoft System Center Configuration Manager Web Console Home Page

The Software Distribution page is the “home” page of Configuration Manager Web Console. From this page you can choose to distribute a new package, perform package administration, and configure distribution settings.

The following selections are available:

Table 19-4 • Configuration Manager Web Console Home Page Options

Option	Description
Distribute a New Package	Click to access the Distribute Package Page where you can select a package to distribute using distribution providers.
Package Administration	Click to access the Package Administration Page , where you can view the status of distributed applications, modify settings of distributed packages, and delete an existing distribution.
Distribution Settings	Click to access the Distribution Settings Page , where you can configure global distribution settings or configure distribution provider settings.

Connect to Configuration Manager Page

This dialog appears when you are attempting to connect to a Configuration Manager Server.

All of the settings configured on this page only apply to the selected package. The default settings (specified on the [Distribution Settings Page](#)) do not change.

Table 19-5 • Connect to Configuration Manager Page Options

Option	Description
Site Server	Enter the machine name on which Configuration Manager Server is installed. (This could be a local or remote machine.) When you make a selection in this field, the Site Code field is populated with a list of sites available on the Configuration Manager Server.
Site Code	Select the Configuration Manager site you want to use for software distribution.
User Name	Enter a valid user name for the Configuration Manager Server you want to connect to, such as: Company\UserName .
Password	Enter the appropriate password.
Next	Click to proceed with the connection.

Distribute Package Page

The Windows Installer .msi packages that you are preparing for distribution to Configuration Manager Server are stored in the AdminStudio Enterprise Server Application Catalog. All of the Windows Installer packages in the open Application Catalog are listed on the Distribute Package page, making them available for distribution. On this page, select the package that you want to distribute to a Configuration Manager Server and click **Distribute**.

This page includes the following information and functionality.

Table 19-6 • Distribute Package Page Options

Option	Description
Application Name	Lists the names of all applications available for distribution. Click the radio button next to the Application Name to select it for distribution. Only one application can be selected at a time.
Company Name	Lists the company affiliated with the listed application.
Distribute	Click to initiate distribution of the selected application.

PDF Selection Page

A Package Definition File (.pdf) is a text file that contains predefined programs and property settings for a package and is used to automate package creation. When you use a package definition file, appropriate fields on the [Configuration Manager Package Configuration Page](#) are populated with the information in the package definition file you selected. A package definition file typically has an .sms file name extension (.pdf in previous versions of Configuration Manager).

This page contains the following fields:

Table 19-7 • PDF Selection Page Options

Option	Description
Distribute Using Package Definition	Select this option if you want to distribute this application using an existing Configuration Manager Package Definition File (.pdf or .sms). When you select this check box, the Use Existing Package Definition option is automatically selected, and the list of existing Package Definitions available on the Configuration Manager Server are shown on the following grid.
Use Existing Package Definition	Select this option to distribute this application using any Package Definitions available on the Configuration Manager Server. When the Distribute using Package Definition check box is selected, this option is automatically selected, and the list of existing Package Definitions available on the Configuration Manager Server are shown on the following grid.

Table 19-7 • PDF Selection Page Options (cont.)

Option	Description
Use Package Definition from the following file	Select this option if you want to distribute this application using a .pdf or .sms file located on the machine where Configuration Manager Console is installed. When you select this option, the list below it is populated with a list of all of the .pdf or .sms files in the following directory: <Configuration Manager Web Console Installation Directory>\SMS Web Console Application\Distribution Providers\SMS\PDFs
Next	Click to proceed to the Configuration Manager Package Configuration Page where appropriate fields are populated with the information in the Package Definition you selected.

Configuration Manager Package Configuration Page

The **Configuration Manager Package Configuration** page, which you use to specify distribution configuration fields that the Configuration Manager Server utilizes in the distribution of the package, consists of the following views:

Table 19-8 • Configuration Manager Package Configuration Page Views

View	Description
Package Settings View	General information about the package, along with data source, data access, distribution settings, and reporting information.
Access Accounts View	Specify which administrators are allowed to distribute this particular package.
Distribution Points View	Specify which servers may act as distribution points for this particular package. A Distribution Point is a site system role that stores software package files so clients can access them during the software distribution process.
Programs View	Specify programs (that will run on clients) to execute the distribution of the application, and the types of clients on which the program can run (defined using directories, operating system versions, etc.).
Advertisements View	Identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory.

Package Settings View

Information on the Package Settings View is presented on the following tabs:

- [General Tab](#)
- [Data Source Tab](#)
- [Data Access Tab](#)
- [Distribution Settings Tab](#)
- [Reporting Tab](#)

Enter and edit information on these tabs and then click **Update** to save your changes.

General Tab

The **General** tab of the **Package Settings View** contains the following options:

Table 19-9 • Package Settings View — General Tab Options

Option	Description
Icon	To change the displayed icon, select a different icon from the list. This list is populated with the files available from: <Configuration Manager Web Console Installation Directory>\SMS Web Console Application\Distribution Providers\SMS\Icons
Name	The name of the package, up to 50 characters. This field is required.
Version	The version number of the software package, up to 32 characters.
Publisher	The name of the software publisher, up to 32 characters.
Language	The language version of the software, up to 32 characters.
Comment	Optional text about the package, such as a description. You can use up to 127 characters.

Data Source Tab

The **Data Source** tab of the **Package Settings View** contains the following options:

Table 19-10 • Package Settings View — Data Source Tab Options

Option	Description
This Package contains source files	This package has source files that are required to run its programs when they are advertised to clients. By default, this check box is cleared. When this check box is cleared, Configuration Manager does not use distribution points for the package.
Source Directory	The network path to the package source files.

Table 19-10 • Package Settings View — Data Source Tab Options (cont.)

Option	Description
Use a compressed copy of the source directory	<p>Configuration Manager will create a compressed version of the source files on the site server. When you specify additional or refresh distribution points for the package, the compressed version will be decompressed and copied to the distribution points, rather than being copied directly from the original source directory.</p> <p>Use this option if the source files might be removed from the specified path (for example, if the source files are on a compact disc). A compressed version of the source directory is always created and used when the package is sent to distribution points in child sites.</p> <p>This field is available only when the This Package contains source files check box is selected.</p>
Always obtain files from source directory	<p>Configuration Manager will obtain package source files (the files needed to run the package programs, such as .exe and .dll files) from the path specified in the Source directory box. Use this option if the source files are likely to remain at the specified path, such as a share on a server.</p> <p>This field is available only when the This package contains source files check box is selected. By default, this option is selected.</p> <div data-bbox="573 1003 609 1045" data-label="Image"> </div> <p>Note • This option is unavailable if the package is a compressed version replicated from a parent site.</p>

Data Access Tab

The **Data Access** tab of the **Package Settings View** contains the following options:

Table 19-11 • Package Settings View — Data Access Tab Options

Property	Description
Access distribution folder through common Configuration Manager package share	<p>Users will access the package through the common Configuration Manager package share (SMSPKGx\$, where x is the drive letter) on distribution points. By default, this option is selected.</p>
Share Distribution folder	<p>The package will be created on distribution points with an administrator-specified share (and path name) that is entered in the Share name text box.</p>


Table 19-11 • Package Settings View — Data Access Tab Options (cont.)

Property	Description
Disconnect users from distribution points	<p>Notifies users before disconnecting them from distribution points when Configuration Manager updates package data. By default, this check box is cleared.</p> <ul style="list-style-type: none"> • Number of retries before disconnecting users—The number of times that Configuration Manager tries to update the package source files before starting to disconnect users who are connected to the distribution point. The default setting is 2; you can specify 0 to 99, inclusive. • User grace period—The number of minutes Configuration Manager should wait after notifying users before disconnecting them from distribution points. The default setting is 5 minutes; you can specify between 0 and 59 minutes, inclusive.

Distribution Settings Tab

The **Distribution Settings** tab of the **Package Settings View** contains the following options:

Table 19-12 • Package Settings View — Distribution Settings Tab Options

Option	Description
Sending priority	<p>The priority of this package when sent to distribution points in child sites. Packages can be sent with High, Medium, or Low priority. The default setting is Medium priority.</p> <ul style="list-style-type: none"> • If a package has High priority, it will be sent before packages with Medium or Low priority. • If a package has Low priority, it will be sent after packages with higher priority settings. • If packages have identical priorities, they will be sent in the order in which they were created in the Configuration Manager Administrator console. <p>Sending priority for a package is carried over in packages sent from a parent site to a child site.</p>
Preferred sender	<p>The sender to be used to send this package to distribution points in other sites. You can specify any sender configured for the site, or No Preference (which uses any available sender). The default setting is No Preference.</p> <div>  </div> <p>Note • To use Courier Sender to distribute packages, you must select it as the preferred sender.</p>

Reporting Tab

The **Reporting** tab of the **Package Settings View** contains the following options:

Table 19-13 • Package Settings View — Reporting Tab Options

Property	Description
Use package properties for status MIF matching	Configuration Manager will use the properties in the General tab for status MIF file matching. By default, this option is selected.
Use these fields for status MIF matching	<p>Configuration Manager will use the following values specified for status MIF file matching:</p> <ul style="list-style-type: none">• MIF file name—The name of the MIF file that contains the package status, up to 50 characters.• Name—The name of the package, up to 50 characters.• Version—The version number of the package, up to 32 characters.• Publisher—The software publisher of the package, up to 32 characters.

Access Accounts View

Use this view to create new users and groups. If you select a user's icon under Access Accounts in the tree, you can review (but not change) that user's permissions and can delete the user.

Table 19-14 • Access Accounts View Options

Option	Description
Domain User Listing	<p>List of all defined users, displaying the following:</p> <ul style="list-style-type: none">• Name—User name using the syntax of Domain\User.• Type—Identifies whether this is a user or a group, and identifies whether they have Windows or Generic access.• Permissions—Lists the user or group's permission.

Table 19-14 • Access Accounts View Options (cont.)

Option	Description
Create New Access Account	<p>To create a new access account, select whether you want to create a Windows User Access Account or a Generic Access Account, and perform the following steps:</p> <p>To create a Windows User Access Account:</p> <ol style="list-style-type: none"> 1. Enter a User Name using the syntax of Domain\User. 2. Under Account Type, select User or Group. 3. Select this user's permission level from the Permissions list. 4. Click Create. <p>To create a Generic Access Account:</p> <ol style="list-style-type: none"> 1. Under Account Type, select Users, Guests or Administrators. 2. Select this generic account's permission level from the Permissions list. 3. Click Create.

Deleting a User or Group

To delete an individual user or group, select the user icon under **Access Accounts** in the tree and click **Delete**.

Distribution Points View

On the Distribution Points View, you can view a list of distribution points available on the Configuration Manager Server and can target them for this package.

A Distribution Point is a site system role that stores software package files so clients can access them during the software distribution process. See [Distributing Software Through Configuration Manager Server](#) for more information.

Table 19-15 • Distribution Points View Options

Option	Description
Distribution Point Listing	<p>List of all defined distribution points, displaying the following:</p> <ul style="list-style-type: none"> • Name—The name of the distribution point. • Site—The name of the site where the distribution point is located (for example, NYC - New York City). • Type—The type of distribution point (for example, Windows NT Server or NetWare Bindery).
Select New Distribution Point	<p>To target a new distribution point(s), select a distribution point(s) from the list and click OK. The new distribution point will now be listed under Distribution Points in the tree.</p>

Programs View

To view a listing of existing programs and to create a new one, use the main **Programs View**.

Table 19-16 • Programs View Options

Option	Description
Program Listing	List of all defined programs, displaying the program name, run time, disk space, and any comments associated with the program.
Create New Program	To create a new program, enter a name in the Name text box and click Create . The new program will now be listed under Programs in the tree. To edit the program settings on the detailed tab view, select the program in the tree.

To edit an existing program, select the program icon under **Programs** in the tree to access the detailed tab views (**General**, **Requirements**, **Environment**, and **Advanced**) where you can edit the program settings.

General Tab

Make edits in these fields and then click **Update** to save your edits. To delete the program, click **Delete**.

Table 19-17 • Programs View — General Tab Options

Option	Description
Icon	To change the displayed icon, select a different icon from the list. This list is populated with the files available from: <Configuration Manager Web Console Installation Directory>\SMS Web Console Application\Distribution Providers\SMS\Icons
Name	The name of the program, up to 50 characters. This field is required and must be unique within a package. This name is used when defining advertisements.
Comment	Optional text about the program, such as a description. You can use up to 127 characters.

Table 19-17 • Programs View — General Tab Options (cont.)




Option	Description
Command Line	<p>The command line for the program, up to 511 characters. This field is required. File name extensions are strongly recommended.</p> <p>Click Browse to navigate to the file. By default, the package source directory appears, if it exists and is accessible. When the program is run on a client, the command line file name will first be searched for within the package.</p> <ul style="list-style-type: none"> • If the file does not exist there, Configuration Manager searches the local Windows directory. • If Configuration Manager cannot find the file, the program fails. • If the file name does not have a file name extension specified, Configuration Manager tries .pif, .com, .exe, and .bat. • If the file name has an extension but is not an executable, Configuration Manager tries to apply a local association. For example, if the command line is readme.gif, Configuration Manager will start the application specified on the client for opening .gif files. For example: <pre>setup.exe /a command.com /c copy Jan98.dat c:\sales\Jan98.dat</pre>  <p>Note • On 16-bit clients, batch files <i>MUST</i> specify the .bat file name extension, or the command line will fail.</p>
Start in	<p>The executable directory for the program, up to 127 characters. This directory can be an absolute path on the client, or a path relative to the distribution point folder that contains the package. This field is optional. For example:</p> <pre>c:\office97 i386</pre>
Run	<p>The program mode. By default, programs run in Normal mode. You can also run a program in Minimized, Maximized, or Hidden mode.</p>

Table 19-17 • Programs View — General Tab Options (cont.)

Option	Description
After Running	<p>The action that occurs after this program is completed successfully.</p> <ul style="list-style-type: none"> • No action required—No restart or logoff is required. This is the default value. • Program restarts computer—The program requires a restart and performs the restart automatically. • Configuration Manager restarts computer—Configuration Manager restarts the computer. • Configuration Manager logs user off—Configuration Manager logs the current user off the computer after the program is completed. Use this option if the program requires that the user log off and then log on again before it can run.  <p>Note • On 16-bit clients, only No action required is supported.</p>  <p>Caution • Both the Configuration Manager restarts computer and the Configuration Manager logs user off actions are done forcefully after providing the user with a grace period. Applications that are running on clients when this occurs will not have an opportunity to save their states or data.</p>

Requirements Tab

Make edits in these fields and then click **Update** to save your edits. To delete the program, click **Delete**.

Table 19-18 • Programs View — Requirements Tab Options



Option	Description
Estimated disk space	A whole number greater than or equal to zero that represents the amount of disk space the software will require when installed on the client. Also, specify the units for the value.
Maximum allowed run time	A whole number greater than zero that represents the time (in minutes) required to run the program on the client.
Notify user if the program runs 15 minutes longer than estimated	<p>Configuration Manager will monitor the run time of the program and warn the user if the program is taking longer to run than the estimated time. By default, this check box is cleared.</p>  <p>Note • This feature is not supported on 16-bit clients.</p>
This program can run on any platform	The program can run on any platform. If this option is selected, Configuration Manager does not check the platform type when advertising the program to clients. By default, this option is selected.

Table 19-18 • Programs View — Requirements Tab Options (cont.)

Option	Description
This program can run only on specified client platforms	<p>The processors, operating systems, and service packs on which this program will run. When this option is selected, at least one platform must also be selected. By default, no platforms are selected. Configuration Manager uses this information when evaluating which clients in a collection are to receive the advertised program.</p>  <p>Note • If you advertise this program to clients in Configuration Manager 1.2 sites, Configuration Manager performs a less in-depth level of platform checking than it does for Configuration Manager 2.0 clients. Platform checking for Configuration Manager 1.2 clients is limited to Windows 3.1, Windows 95, x86 Windows NT, Alpha Windows NT, and Macintosh.</p>
Additional Requirements	<p>Other information for clients, up to 127 characters. This field is optional. For example, you can notify users that they should shut down all other applications before running this program.</p>


Environment Tab

The **Environment** tab of the **Programs View** contains the following options:

Table 19-19 • Programs View — Environment Tab Options

Option	Description
Program can run	<p>Specify whether the program requires that a user be logged on to the client computer to run:</p> <ul style="list-style-type: none"> • Only when a user is logged on—Prevents the program from running if no user is logged on to the computer. This is the default setting. Use this option for clients that do not run a Windows NT operating system. • Only when no user is logged on—Prevents the program from running until the user logs off the computer. This option is valid only for Windows NT clients. This option forces the program to run under the local Administrator account on the client. If a user logs on while the installation is running, the program continues to run. • Whether or not a user is logged on—Enables the program to run with or without user interaction. This option is valid only for Windows NT clients. This option forces the program to run under the local Administrator account on the client.
User input required	<p>Requires that the user interact with the program when it runs. This check box is selected when Program can run is set to Only when a user is logged on. If the program does not require user input, clear this check box.</p>

Table 19-19 • Programs View — Environment Tab Options (cont.)

Option	Description
Run mode	<p>Select one of the following run modes:</p> <ul style="list-style-type: none"> • Run with user's rights • Run with administrative rights <p>To specify that the program requires administrative privileges on the computer in order to run, select Run with administrative rights. This option is available when Program can run is set to Only when a user is logged on. By default, this check box is cleared. This option is valid only for Windows NT clients.</p> <p>The Run with administrative rights option forces the program to run under the local Administrator account on the client.</p>
Use software installation account	<p>Run the program under an administrator-specified account (the Configuration Manager Windows NT Client Software Installation Account) on Windows NT clients. This check box is available only when Program can run is set to Only when no user is logged on or Whether or not a user is logged on. By default, this check box is cleared.</p> <p>This option should be used only when the program requires access to network resources other than the package files on the distribution point.</p>
Drive mode	<p>Select the drive mode used for network connections to the distribution points.</p> <ul style="list-style-type: none"> • Runs with UNC name—Enables the program to run with a universal naming convention (UNC) name. This is the default setting. • Requires drive letter—Indicates that the program requires a drive letter to fully qualify its location, but Configuration Manager can use any available drive letter on the client. • Requires specific drive letter—Indicates that the program requires the drive letter you specify (A to Z) to fully qualify its location (for example, H). If the specified drive letter is already used on a client, the program will not run.
Reconnect to Distribution point at logon	<p>The computer will reconnect to the drive when the user logs on. By default, this check box is cleared.</p>  <hr/> <p>Note • This feature is not supported on 16-bit clients.</p>

Advanced Tab

The **Advanced** tab of the **Programs View** contains the following options:

Table 19-20 • Programs View — Advanced Tab Options






Option	Description
Run another program first	<p>This program requires another program (in another package that is targeted for distribution using Configuration Manager Web Console or in the current package) to run before it. By default, this check box is cleared.</p>  <p>Note • This feature is not supported on 16-bit clients.</p>  <p>Note • The program you specify to run first does not need to be advertised separately.</p>
Package	The package that contains the program that must run before this program.
Program	<p>The program that must run first.</p>  <p>Note • If the depended-upon program fails on a client, the depending program will not run. Also, if the program that runs first has been run successfully, Configuration Manager will not automatically re-run it, even if the depending program is on a recurring schedule.</p>
When this program is assigned to a computer	<p>When Program can run (in the Environment tab) is set to Only when a user is logged on, you can specify how the program is assigned to users.</p> <ul style="list-style-type: none"> • Run once for first user who logs on—The program should run only once on the computer. This is the default setting. For example, a virus checker only needs to run once on the client. • Run once for every user who logs on—The program should run once for each new user who logs on.  <p>Note • On 16-bit clients, advertised programs will run only for the first user who logs on.</p>

Table 19-20 • Programs View — Advanced Tab Options (cont.)

Option	Description
Remove software when it is no longer advertised	<p>Indicates that Configuration Manager will remove this program from client computers when it is no longer advertised to those clients. This option is mainly useful for shared network applications and is available only to programs that register with Add/Remove Programs in Control Panel.</p> <p>To use this option, enter the name of the Uninstall Registry Key the program creates in the following registry key:</p> <p>HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\ CurrentVersion\Uninstall</p> <p>After all advertisements for the program are revoked from the client, Configuration Manager looks up the program's uninstall command line in the UninstallString or QuietUninstallString property in this registry key. By default, this check box is cleared.</p> <p>An advertisement is considered revoked from a client when any of the following are true:</p> <ul style="list-style-type: none"> • The client is no longer a member of the collection specified in the advertisement. • The advertisement is deleted. • The advertisement expires. • Uninstall registry key • The registry key that contains the uninstall command for this program. You can use up to 127 characters.  <p>Note • This feature is not supported on 16-bit clients.</p>
Disable this program on computers where it is advertised	<p>Temporarily disables all advertisements that contain this program. The program will be removed from the list available for users to run and will not be run via assignment until it is re-enabled. By default, this check box is cleared.</p>

Advertisements View

On the Advertisements View, you identify the collection of users to receive this package, and specify when the package will be available for distribution and whether it is optional or mandatory. You can view a list of existing advertisements and create new advertisements.

Table 19-21 • Advertisements View Options

Option	Description
Advertisement Listing	<p>List of all existing advertisements, displaying the following:</p> <ul style="list-style-type: none"> • Name—Name of the advertisement • Program—The program to run on the client. • Collection—The target collection of computers, users, or user groups that will receive the program. • Available After—Date after which the advertisement will be available • Expires After—Expiration date of advertisement • Advertisement ID—Number uniquely identifying the advertisement • Status—Advertisement's status
Create New Advertisement	To create a new advertisement, enter a name in the Name text box and click Create. The new advertisement will now be listed under Advertisements in the tree.

If you select an existing advertisement under Advertisements in the tree, you can edit it on the detailed tab view (**General** and **Schedule**).


General Tab

To edit or delete an advertisement, select the advertisement icon under Advertisements in the tree. The General tab appears. Make edits in these fields and then click Update to save your edits. To delete the advertisement, click Delete.

Table 19-22 • Advertisements View — General Tab Options

Option	Description
Name	The name of the advertisement, up to 50 characters. This field is required.
Comment	Optional text about the advertisement, such as a description. You can use up to 127 characters.
Program	The program (within the current package) to advertise to clients. This field is required.

Table 19-22 • Advertisements View — General Tab Options (cont.)

Option	Description
Collection	<p>Select the existing collection that will receive the advertisement. Click Browse to open the Select a Configuration Manager Collection Dialog Box, where you can select the Configuration Manager collection that you want to use in this Advertisement definition. This field is required.</p>  <p>Note • Configuration Manager will not advertise a program to 16-bit clients based on user groups or user accounts, but this is supported for 32-bit clients.</p>
Include members of subcollections	<p>Specifies that the program also be advertised to clients and users in the specified collection's subcollections. By default, this check box is selected.</p>

Schedule Tab

Make edits in these fields and then click **Update** to save your edits. To delete the advertisement, click **Delete**.

Table 19-23 • Advertisements View — Schedule Tab Options



Option	Description
Advertisement start time	<p>The date and time when the program is advertised and available to run on clients. By default, the current date and time is specified.</p> <p>Greenwich Mean Time—Sets the time independent of time zones. That is, if you set the time to 1:00 P.M. and select this option, the program will be advertised on all clients simultaneously—1:00 P.M. for clients in the Greenwich mean time zone, 10:00 A.M. for clients in the Eastern time zone, and 7:00 A.M. for clients in the Pacific time zone, and so on. This check box is cleared by default, which enables you to specify a relative time for advertising the program. Configuration Manager will treat the specified time as a local time. In this case, if the time is set to 1:00 P.M., the program will be advertised on clients at 1:00 P.M. in each time zone.</p>
Mandatory assignments (Assign immediately after this event)	<p>If you want to ensure that the program runs on all clients (users cannot choose not to run the program), select:</p> <ul style="list-style-type: none"> • As soon as possible • Logon • Logoff  <p>Note • Recurring assignments are not supported on 16-bit clients. Programs advertised to 16-bit clients run only once on those clients.</p>

Table 19-23 • Advertisements View — Schedule Tab Options (cont.)

Option	Description
Assignments are not mandatory over slow links	Suspends assignments for clients on a slow link. By default, this check box is selected. (This option is available only if at least one assignment is defined.)
Allow users to run the program independently of assignments	<p>Enables users to run the program independent of when it is assigned. By default, this check box is cleared. (This option is available only if at least one assignment is defined.)</p> <p>Typically, only non-assigned programs are visible to users in the Configuration Manager client interface. Use this option to display an assigned program in the client interface. For example, to require users to upgrade their Microsoft Office software, you specify an assignment in the advertisement properties. But if you want to enable users to run the installation program before it is scheduled to run automatically, select this option so that the assigned program appears in Advertised Programs Wizard in the Control Panel on client computers.</p>
Advertisement will expire	<p>Advertises the program to clients only for a limited time. By default, this option is cleared. This feature is not supported on 16-bit clients; the program appears until it runs on the client.</p>  <p>Note • When a program expires, it no longer appears in Advertised Programs Monitor or Advertised Programs Wizard in Control Panel on client computers. To specify the cutoff date and time, click the arrows next to the date and time. The default values are six months from the current date and time.</p> <p>Greenwich Mean Time—Set the time independent of time zones. That is, if you set the time to 1:00 P.M. and select this option, the program will expire at all computers simultaneously—1:00 P.M. for clients in the Greenwich mean time zone, 10:00 A.M. for clients in the Eastern time zone, and 7:00 A.M. for clients in the Pacific time zone, and so on. This check box is cleared by default, which enables you to specify a relative time. Configuration Manager will treat the specified time as a local time. In this case, if the time is set to 1:00 P.M., the program will expire at 1:00 P.M. on clients in each time zone.</p>
Priority	The advertisement's priority when sent to child sites: High, Medium, or Low. The default priority is Medium.

Package Summary Page

This page, which lists a summary of the information in the .pdf file that you are using for Configuration Manager distribution, appears after you have reviewed the .pdf file on the [Configuration Manager Package Configuration Page](#).

Click **Commit Changes to Configuration Manager** to save the package on the Configuration Manager Server.

A message will appear informing you that the package changes were successfully saved to Configuration Manager.

Select a Configuration Manager Collection Dialog Box

This dialog box, which appears when you click **Browse** next to the **Collection** field on the **General** tab of the [Advertisements View](#) of the [Configuration Manager Package Configuration Page](#), lists all defined collections on the Configuration Manager Server.

Select the collection that you want to use for the Advertisement definition, and click **OK**.

Distribution Settings Page

On the **Distribution Settings** page, which opens when you click **Distribution Settings** on the **Configuration Manager Web Console** Home Page, you set the default Configuration Manager Server connection settings.

Table 19-24 • Distribution Settings Page Options

Option	Description
Site Server Name	Enter the machine name on which Configuration Manager Server is installed. (This could be a local or remote machine.) When you make a selection in this field, the Site Code field is populated with a list of sites available on the Configuration Manager Server.
Site Code	Select the Configuration Manager site you want to use for software distribution.
User Name	Enter the user name appropriate for the Configuration Manager Site you want to connect to.
Update	Click to save your entries.



Note • The values entered on this page will be used as the default values for any new package you target for distribution. However, you can change these values while configuring the package.

Package Administration Page

On this page you can view the status of distributed applications, modify settings of distributed packages, and delete an existing distribution.

Table 19-25 • Package Administration Page Options

Option	Description
Configuration Manager Package ID	Package ID assigned to this package by Configuration Manager.
Package Name	Name of the package.
Version	Version of the package.
Language	Language that the package was written for.
Description	Description of package.
Status	Lists the distribution status of the package: Distributed or Not Distributed .
Application Name	Name of application.
Company Name	Manufacturer of application.
Modify Settings	Click to modify the selected package's distribution settings. After you connect to the appropriate Configuration Manager Server, the Configuration Manager Package Configuration Page appears, where you can modify the distribution settings.
View Status	Click to view the status of a distributed application. Configuration Manager Web Console connects to the Configuration Manager Server, retrieves the status of that application, and displays that status as a text message.
Delete Distribution	Click to delete the selected application from Configuration Manager Web Console and/or the Configuration Manager Server.

Package Status Page

The Package Status page opens when you select a Distribution Package on the [Package Administration Page](#), and click **View Status**.

A summary of this distributed package status is displayed in two main sections: **Package Status** and **Advertisement Status**.

Package Status

The Package Status section includes the following information:

Table 19-26 • Package Status Page — Package Status Information

Option	Description
Site	Server location where package was distributed.
Distribution Point	Distribution point where this package was distributed. A distribution point is a site system role that stores software package files so clients can access them during the software distribution process.
State	Identifies the state of the source files on the distribution point. Values are: <ul style="list-style-type: none"> • INSTALLED (0) • INSTALL_PENDING (1) • INSTALL_RETRYING (2) • INSTALL_FAILED (3) • REMOVAL_PENDING (4) • REMOVAL_RETRYING (5) • REMOVAL_FAILED (6)
Last Copied	Date and time (GMT) when the package source files were last successfully copied to the distribution point.
Source Version	The number of times an administrator has updated the package source files and requested an update to distribution points (specified on the Data Source tab of the Package Properties dialog box).
Targeted	The number of distribution points this package was sent to.
Installed	The number of distribution points that received this package.
Retrying	The number of distribution points that are having problems receiving the package, but that Configuration Manager is still attempting to deliver to.
Failed	The number of distribution points that failed to receive the package. Failure means Configuration Manager tried to deliver the package several times, but could not do so successfully.
Summary Date	Date and time (GMT) when a change in package status for the sites was most recently reported.
Path	NAL path to the package's source files.

Advertisement Status

The **Advertisement Status** section includes the following information

Table 19-27 • Package Status Page — Advertisement Status Information

Option	Description
Advertisement Name	Name of the advertisement.
Site	Server location where package was distributed.
Target Collection	Name of the collection to which this advertisement is advertised
Program	Name of the program (related to PackageID) that this advertisement advertises.
Available After	A program available to members of a collection after this time.
Expires After	A program will no longer be available to members of a collection after this time.
Received	Total number of users, user groups, or client computers that are reporting successful receipt of the advertisement.
Failures	Total number of users, user groups, or client computers that experienced an error in processing the advertisement or its associated program, or that attempted to run the advertised program but failed before the program could be started.
Programs Started	Total number of users, user groups, or client computers in the site that were able to successfully start running the advertised program.
Program Errors	Total number of users, user groups, or client computers that reported errors while running the advertised program. A program is considered in error when it produces either: <ol style="list-style-type: none"> 1. A non-zero exit code, or 2. An install-status MIF file with a failure-status attribute. This file, if present, will override an exit code.
Program Success	Total number of users, user groups, or client computers, or both, that are reporting that the advertisement ran successfully. A program is considered successful when it produces either: <ol style="list-style-type: none"> 1. A exit code of zero, or 2. An install-status MIF file with a success-status attribute. This file, if present, will override an exit code.
Advertisement ID	Unique identifier that Configuration Manager assigns to each advertisement.

Confirm Delete Page

The **Confirm Delete** page opens when you select a package on the [Package Administration Page](#), and click **Delete Distribution**.

The messages shown on this page depend upon the status of the selected package: **Distributed** or **Not Distributed**:

- For both package status types, you are prompted to confirm that you want to delete the package from the Application Catalog.
- If you are deleting a **Distributed** package, you are also prompted to confirm that you want to delete this package from Configuration Manager.

Select **Yes** or **No** to indicate your deletion preferences:

Table 19-28 • Package Deletion Options

Package Status	Delete from Catalog?	Delete from Configuration Manager?
Not Distributed	Choose Yes to confirm the deletion from the Application Catalog or choose No to decline the deletion. If you choose Yes , the package will be deleted from the Application Catalog, along with all of the package configuration information that you entered on the Configuration Manager Package Configuration page.	N/A
Distributed	Choose Yes to confirm the deletion from the Application Catalog, or choose No to decline the deletion. If you delete a package from Configuration Manager but do not delete it from the Application Catalog, all of the package configuration information that you entered on the Configuration Manager Package Configuration page is saved, making it easy to distribute this package again in the future.	Choose Yes to confirm the deletion from Configuration Manager, or choose No to decline the deletion. Unless you are certain that you will never want to distribute this package again, it is recommended that if you delete a package from Configuration Manager, you do not delete it from the Application Catalog.

Click **OK** to confirm your selections.



Caution • Deleting a distributed package from Configuration Manager will also delete its programs and any advertisements of the programs. If the package has source files, they will be removed from distribution points. If there are access accounts for this package, they will be deleted. Also, Configuration Manager Administrators' security rights to the package will be deleted.



Caution • When you delete a package from the Application Catalog, all package configuration information is lost, even if you do not delete it from Configuration Manager. If you delete a package from the Application Catalog but not from Configuration Manager, the package would still be available on the Configuration Manager Server, but all connection between the Application Catalog and Configuration Manager is lost. Should you want to distribute that package again in the future, you would have to re-import the application into the Application Catalog, and then re-enter package configuration information on the [Configuration Manager Package Configuration Page](#).

Distribute New Package Page



Edition • This feature is included with AdminStudio Standard Edition only.

In the **Application Name** field, enter the name of the application that you want to distribute to a Configuration Manager Server, and click **Distribute** to continue.

Chapter 19: Distributing Packages Using Configuration Manager Web Console

Configuration Manager Web Console Reference

Analyzing Installations Prior to Deployment



Edition • *Predeployment Test is included with AdminStudio Professional and Enterprise Editions.*

The Predeployment Test tool allows system administrators to determine if a Windows Installer .msi package will succeed or fail when it is installed in production. Predeployment Test finds issues with disk space, install conditions, and will find conflicts with Windows installer packages already installed on the system.

The Predeployment Test user documentation is organized into the following sections:

Table 20-1 • Predeployment Test User Documentation

Section	Description
About the Predeployment Test Tool	This section provides an overview of the Predeployment Test tool and explains how it works.
Configuring the Predeployment Test Environment	This section explains how to configure the Predeployment Test Results Web Site and how to set the Predeployment Test Results Web site addresses.
Performing Predeployment Analysis	This section provides step-by-step procedures explaining how to generate and distribute a test-ready Windows Installer package (a package that does not make any system changes when it is deployed), and how to generate test reports.
Viewing Predeployment Test Results	This section describes each of the reports available on the Predeployment Test Results Web site.
Predeployment Test Reference	Every view, dialog box, and Wizard available in the Predeployment Test tool is discussed in this section.

About the Predeployment Test Tool

The Predeployment Test tool allows system administrators to determine if a Windows Installer .msi package will succeed or fail when it is installed in production. Predeployment Test finds issues with disk space, install conditions, and will find conflicts with Windows Installer packages already installed on the system.

This section includes the following topics:

- [How the Predeployment Test Tool Works](#)
- [Predeployment Test Tool Components](#)
- [Overview of Predeployment Test Tool Results](#)



Caution • Predeployment Test features cannot be installed on 64 bit machines because Microsoft® .NET Framework 2.0 does not support 64 bit machines.

How the Predeployment Test Tool Works

When using the Predeployment Test Tool to determine if a Windows Installer package will succeed or fail when it is installed in production, you perform the following steps:

- **Create test-ready packages**—Use the [Predeployment Test Preparation Wizard](#) to convert an original Windows Installer .msi package into a “test-ready” package which incorporates custom actions. A test-ready .msi package does not make any system changes when it is installed; it is used to collect data on whether that package would succeed or fail if it was installed on a particular machine. Using the wizard, you select the package to test and you specify deployment and conflict tests that you want to execute on the deployment machines.
- **Install test-ready packages**—You install the test-ready packages in your enterprise. When a test-ready .msi package is deployed, Type 1 .msi custom actions are executed. Each custom action performs a test and sends the results to a Web service, and the Web service saves those results into a database.
- **Analyze results**—You can then view the compiled test results in a Web browser.

Predeployment Test Tool Components

The Predeployment Test Tool consists of five components:

- **Predeployment Test Preparation Wizard**—This wizard-based tool converts the original .msi package into a “test-ready” .msi package, a package that does not make any system changes when it is deployed.
- **Custom Actions**—Type 1 .msi custom actions are executed when the “test-ready” .msi package is deployed. Each custom action performs a test and sends the results to the .NET Web service.
- **.NET Web Service**—This .NET Web service accepts the results from the custom action execution and saves them into database tables in the Application Catalog.

- **Database Tables**—Test results from the custom action execution are stored in database tables in the Application Catalog. The .NET Web service writes into this database.
- **Predeployment Test Results**—The Predeployment Test tool accesses the saved test results in the database tables. You can click a link on the **Predeployment Test Start Page** to view the test results in a Web browser.

Overview of Predeployment Test Tool Results

When each test-ready package is deployed, test results are added to its corresponding job on the Predeployment Test Results Web site and those results are displayed in the following reports:

Table 20-2 • Predeployment Test Results

Report Name	Description
Predeployment Test Results By Job Report	Lists all of the “test-ready” package jobs that have been run. Each time a test-ready package is installed on a machine in your network, test results are added to its corresponding job on the Predeployment Test Results Web site.
Job Summary Report	Lists the total number of times the test-ready package was run, and the number of times it Passed, Failed, and was Incomplete.
Test Result Summary Report	Lists the all the computers that this job was run on (All Results), or all the computers that had either: Passed, Failed, or Incomplete results.
Job Details Report	Lists all of the deployment and conflict tests that were run on this machine for this job.
Test Details Report	Lists detailed information on errors that were generated during a Deployment Test.
File/Registry Conflicts Report	Lists the results of File Conflict checks (ACE07, ACE23) and Registry Conflict checks (ACE10, ACE24).
Predeployment Test Results By Machine Report	Lists all of the machines in the network where “test-ready” packages have been installed.
Machine Result Summary Report	Lists all of the test-ready package jobs run on a specific machine.

Configuring the Predeployment Test Environment

To configure the Predeployment Test environment, you specify the following:

- **Predeployment Test Results Web Site**—Web site address where the Predeployment Test Results can be viewed, and the Application Catalog that stores the reports.
- **Predeployment Test Web Service**—Web site address of the Predeployment Test Web Service application, and the Application Catalog that stores the data generated when a test-ready application is installed.

This section includes the following topics:

- [Configuring the Predeployment Test Results Web Site](#)
- [Configuring the Predeployment Test Web Service](#)

Configuring the Predeployment Test Results Web Site

To configure the Predeployment Test Results Web site, open the [Options Dialog Box](#) and specify the following information:

- **Web Address**—Enter the URL of the Predeployment Test Results Web site. See [Setting the Predeployment Test Results Web Site Address](#).
- **Application Catalog**—Configure a connection to an Application Catalog to store the Predeployment Test results. You will connect to this database to view the reports. [Configuring the Predeployment Test Results Web Site Application Catalog](#)



Note • To use a Microsoft Access Application Catalog database on the Predeployment Test Results Web site, the database file must have ASP.NET permissions assigned to it. Files in the AdminStudio Shared directory inherit ASP.NET permissions from that directory. If you want to open a Microsoft Access database that is in another directory, you must assign ASP.NET permissions to that database file or the directory that it is in. See [Assigning ASP.NET Permissions to Files or Folders](#).

Setting the Predeployment Test Results Web Site Address



Task: *To configure the Predeployment Test Results Web site:*

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. On the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. In the **URL** field of the **Predeployment Test Results Web Site** area, enter the Web address of the Predeployment Test Results Web site.

When Predeployment Test is installed, the default Predeployment Test Results URL is:

`http://IPADDRESS/Predeploytestreports/default.aspx`

where IPADDRESS is the Predeployment Test Service virtual directory address.

4. Click **OK**.

Configuring the Predeployment Test Results Web Site Application Catalog



Task:

To configure the Predeployment Test Results Web Site Application Catalog:

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. On the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. To select the Application Catalog that will store the reports, click **Configure Application Catalog** in the **Predeployment Test Results Web Site** area. The **Connection to New Catalog** page opens.
4. To configure a **Standalone** Application Catalog, perform the following steps:
 - a. Open the **Standalone** tab.
 - b. Enter the name of the **Server** to which you want to connect.
 - c. For **How should SQL Server verify the authenticity of login ID?**, select one of the following options:
 - **With Windows NT authentication using the Network Login ID**—Use Windows network authentication when connecting to this database.
 - **With Database Server authentication using Login ID and password entered by user**—Use database server login identification for authentication when connecting to this database.
 - d. If you choose **Database Server authentication**, enter the **Login ID** and **Password**.
 - e. In the **Catalog Name** field, enter the name of an Application Catalog database available on the Microsoft SQL Server database.
 - f. Click **Save** to set the selected database as the Application Catalog for the Predeployment Test Results Web site.
5. To configure the **AdminStudio Enterprise Server** Application Catalog, perform the following steps:
 - a. Open the **Enterprise Server** tab. The address of the AdminStudio Enterprise Server is listed.
 - **If the AdminStudio Enterprise Server has been configured**, this address will be a URL to a running instance of an AdminStudio Enterprise Server.
 - **If the AdminStudio Enterprise Server has not yet been configured**, this address will be the default value of `http://localhost`.
 - b. **If the AdminStudio Enterprise Server site has not yet been configured**, or if the URL is incorrect, click the address link to open the **Select AdminStudio Enterprise Server URL** page, enter the correct URL, and click **Save**.

- c. On the **Enterprise Server** tab, enter a valid **User Name** and **Password** for the specified AdminStudio Enterprise Server.
- d. Click **Login**.

Configuring the Predeployment Test Web Service

To configure the Predeployment Test Web Service, open the [Options Dialog Box](#) and specify the following information:

- **Web Address**—Enter the URL of Predeployment Test Web Service. See [Setting the Predeployment Test Web Service Address](#).
- **Application Catalog**—Configure a connection to an Application Catalog to receive and store the test result data generated when test-ready packages are installed. See [Configuring the Predeployment Test Web Service Application Catalog](#).



Note • To use a Microsoft Access Application Catalog database on the Predeployment Test Results Web site, the database file must have ASP.NET permissions assigned to it. Files in the AdminStudio Shared directory inherit ASP.NET permissions from that directory. If you want to open a Microsoft Access database that is in another directory, you must assign ASP.NET permissions to that database file or the directory that it is in. See [Assigning ASP.NET Permissions to Files or Folders](#).

Setting the Predeployment Test Web Service Address



Task: **To set the Predeployment Test Web Service address:**

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. On the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. In the **URL** field of the **Predeployment Test Web Service** area, enter the URL of the Predeployment Test Web Service.

When Predeployment Test is installed, the default Predeployment Test Web Service URL is:

`http://IPADDRESS/PreDeploytestservice/predeploytestservice.aspx`

where IPADDRESS is the Predeployment Test Service virtual directory address.

4. Click **OK**.

Configuring the Predeployment Test Web Service Application Catalog



Task:

To configure the Predeployment Test Web Service Application Catalog:

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. On the **Tools** menu, select **Options**. The **Options** dialog box opens.
3. To select the Application Catalog that will receive and store the test result data generated when test-ready packages are installed, click **Configure Application Catalog** in the **Predeployment Test Web Service** area. The **Configure Application Catalog for Predeployment Web Service** page opens.
4. To configure a **Standalone** Application Catalog, perform the following steps:
 - a. Open the **Standalone** tab.
 - b. Enter the name of the **Server** to which you want to connect.
 - c. For **How should SQL Server verify the authenticity of login ID?**, select one of the following options:
 - **With Windows NT authentication using the Network Login ID**—Use Windows network authentication when connecting to this database.
 - **With Database Server authentication using Login ID and password entered by user**—Use database server login identification for authentication when connecting to this database.
 - d. If you choose **Database Server authentication**, enter the **Login ID** and **Password**.
 - e. In the **Catalog Name** field, enter the name of an Application Catalog database available on the Microsoft SQL Server database.
 - f. Click **Save**.
5. To configure the AdminStudio Enterprise Server Application Catalog, perform the following steps:
 - a. Open the **Enterprise Server** tab. The address of the AdminStudio Enterprise Server is listed.
 - **If the AdminStudio Enterprise Server has been configured**, this address will be a URL to a running instance of an AdminStudio Enterprise Server.
 - **If the AdminStudio Enterprise Server has not yet been configured**, this address will be the default value of `http://localhost`.
 - b. If the AdminStudio Enterprise Server site has not yet been configured, or if the URL is incorrect, click the address link to open the **Select AdminStudio Enterprise Server URL** page, enter the correct URL, and click **Save**.
 - c. On the **Enterprise Server** tab, enter a valid **User Name** and **Password** for the specified AdminStudio Enterprise Server.
 - d. Click **Login**.



Caution • Predeployment Test features cannot be installed on 64 bit machines because Microsoft® .NET Framework 2.0 does not support 64 bit machines.

Performing Predeployment Analysis

To analyze a package prior to deployment using the Predeployment Test tool, you create and distribute a test-ready Windows Installer package. You can then view the test results on the Predeployment Test Results Web site.

- [Creating a Test-Ready Windows Installer Package](#)
- [Setting Predeployment Test Command Line Parameters](#)
- [Distributing a Test-Ready Windows Installer Package](#)
- [Viewing Predeployment Test Results](#)
- [Managing Job Data](#)
- [Setting Predeployment Test Command Line Parameters](#)

Creating a Test-Ready Windows Installer Package

You use the **Predeployment Test Preparation Wizard** to convert an original Windows Installer .msi package into a “test-ready” .msi package, a package that does not make any system changes when it is deployed.

This test-ready .msi package incorporates Type 1 .msi custom actions which are executed when the package is deployed. Each custom action performs a test and sends the results to a Web service, and the Web service saves those results into a database. You can then view the test results in a Web browser.



Task: *To create a test-ready Windows Installer package:*

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. Click the **Predeployment Test Preparation Wizard** link. The **Welcome Panel** of the **Predeployment Test Preparation Wizard** opens.
3. Click **Next**. The **Windows Installer File Selection Panel** opens.
4. In the **Source .MSI Package** field, select the Windows Installer .msi package that you want to test with the Predeployment Test tool—the .msi package that is being prepared for deployment.

If you launched the Predeployment Test Preparation Wizard from the Application Manager by selecting a package and selecting **Distribute Package** from the context menu, the name in the **Source .MSI Package** field is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:

- **Not in the Software Repository**—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click **Browse** and select a different package.
 - **In the Software Repository**—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed.
5. In the **Target .MSI Package** field, specify the name and location of the test-ready .msi package that will be generated at the end of this wizard.

This field is automatically populated with the name of the package specified in the **Source .MSI Package** field with _analysis appended to the end of the name. For example, if WinZip.msi is selected as the **Source .MSI Package**, the **Target .MSI Package** will automatically be named WinZip_Analysis.msi. You can edit the name and the location of this file, if desired.

If you launched the Predeployment Test Preparation Wizard from the Application Manager by selecting a package and selecting **Distribute Package** from the context menu, and the package is not in the Software Repository, the full name and path of the Target .MSI Package is displayed, but if the package is in the Software Repository, only the name of the Target .MSI Package is displayed (not the full path).

6. Click **Next**. The **Web Service URL Panel** opens.
7. In the **Web Service URL** field, enter the HTTP Web Service URL where you want to post the Predeployment Test results. This field automatically displays the Web Service URL that was entered the last time the wizard was run. By default, this field is set to:

`http://serveraddress/PreDeployTestService/PreDeployTestService.asmx`

8. In the **Job Identifier** field, enter a title to identify the Predeployment Test results in the analysis reports. By default, this field is populated with the package name.
9. Click **Next**. The **Select Application Catalog Panel** opens.
10. Select the Application Catalog that will store the test results. Choose one of the following options:
- **Send test results to the Application Catalog registered with the Web Service**—Select this option to send the Predeployment Test results to the Application Catalog registered with the Web Service. That Application Catalog's Server and Catalog names are listed
 - **Send test results to an alternate Application Catalog**—Click **Browse** to open the **Select Application Catalog** dialog box and select an Application Catalog to receive the test results.
11. Click **Next**. The **Deployment & Conflict Test Selection Panel** opens.
12. Select one or more deployment and conflict tests that you want to execute on the deployment machines when the test-ready package is deployed. Selecting a folder automatically selects all tests in that folder. Selecting **Predeployment Tests** automatically selects all deployment and conflict tests in all folders.

The selections made on this panel are saved in an XML file (PDTests.xml in the **Support** subfolder of the AdminStudio installation folder) so that the same test selections are displayed the next time the wizard is run.

13. Click **Next**. The **Review Panel** opens.
14. Review the settings you specified in this wizard, and then click **Start** to initiate the generation of the test-ready .msi package. The **Progress Panel** opens, and indicates the progress of the testing.

To edit a setting before initiating testing, click **Back** to return to previous panels.

15. When testing is complete, the **Next** button is activated. Click **Next** to proceed to the **Summary Panel**.
16. The **Summary Panel** reports that the test-ready .msi package was successfully generated, and lists a summary of the package settings. Perform one of the following options:
 - a. **To exit the wizard without launching distribution** of the test-ready .msi package, click **Finish**. Then to distribute the test-ready package at a later time, launch the **Distribution Wizard**, select the test-ready .msi package on the **Installation Package Information Panel**, and proceed with distribution.
 - b. **To automatically launch the Distribution Wizard** so that you can distribute this test-ready .msi package immediately, select the **Distribute the test-ready Windows Installer .msi package now** option and click **Finish**. The **Distribution Wizard** is launched with the generated test-ready .msi package automatically selected as the distribution source. You can then use the distribution method of your choice to distribute the test-ready package.

Setting Predeployment Test Command Line Parameters

This section includes the following topics:

- [Command Line Parameters for the Predeployment Test Preparation Wizard](#)
- [Command Line Arguments for Distributing Test-Ready Packages](#)

Command Line Parameters for the Predeployment Test Preparation Wizard

The Predeployment Test tool supports command-line parameters so that all the information entered in the **Predeployment Test Preparation Wizard** can also be provided via command line. The Predeployment Test Preparation Wizard works in silent mode when the following command line parameters are provided:

Table 20-3 • Predeployment Test Command Line Parameters

Parameter	Description
A	Application Catalog to use
S	Source Windows Installer .msi package file
T	Target test-ready .Windows Installer msi package file that will be generated
W	Web service URL
I	Test identifier
C	Comma-separated list of conflict tests to run
D	Comma-separated list of deployment tests to run

The following is an example of how to use these command line parameters:

```
PreDeploymentWizard.exe -S"c:\temp\my.msi" -T"c:\temp\My_Analysis.msi" -W"http://localhost/MyWebService.asmx"
```

```
PreDeploymentWizard.exe -C"File,Registry" -  
D"AppSearch,LaunchCondition,FileWrite,RegistryPermissions" -I"Test1"
```

Command Line Argument to Specify an Application Catalog

You can use the A command line argument to specify the Application Catalog that the Predeployment Test Preparation Wizard should use:

-A ConnectionString

When -A is not specified, the Wizard will always use the Web Service catalog. Using -A implicitly enables you to specify an alternate Application Catalog for the Wizard to use.

The connection string must be the complete OLE DB connection string including provider information.

Command Line Arguments for Distributing Test-Ready Packages

All test service settings are configurable from the command line when running a test-ready package. Here is a list of installer properties and their affect on the running package.

Table 20-4 • Command Line Arguments for Test-Ready Package

Property	Description
PD_ALTERNATE_CATALOG_PROVIDER	Property defining the database vendor for target Application Catalog. This property is not used unless the following property is defined: PD_ALTERNATE_CATALOG_CXN_STRING
PD_ALTERNATE_CATALOG_CXN_STRING	Connection string for an alternate Application Catalog. When this property is defined, the Web Server is forced to use an alternate Application Catalog. This property must contain the full connection string required to connect to the application catalog.

Distributing a Test-Ready Windows Installer Package

On the last panel of the **Predeployment Test Preparation Wizard**, the **Summary Panel**, you can select an option to automatically launch the **Distribution Wizard** so that you can distribute this test-ready .msi package immediately.



Task:

To distribute a test-ready Windows Installer .msi package:

1. Generate a test-ready Windows Installer .msi package, as described in **Creating a Test-Ready Windows Installer Package**.
2. On the **Summary Panel** of the **Predeployment Test Preparation Wizard**, select the **Distribute the test-ready Windows Installer .msi package now** option and click **Finish**.

The **Distribution Wizard** will launch with the generated test-ready .msi package automatically selected as the distribution source. You can then use the distribution method of your choice to distribute the test-ready package.



Note • If you have already completed the Predeployment Test Preparation Wizard and did not select the **Distribute the test-ready Windows Installer .msi package now** option on the **Summary Panel**, launch the Distribution Wizard and select the test-ready .msi package on the **Installation Package Information Panel**, and proceed with distribution.

Running Diagnostic Tests When Distributing Test-Ready Packages

You can use the Predeployment Test Diagnostic Tool to run diagnostic tests on the critical parts of the predeployment reporting framework.

Diagnostic Tests Performed

The Diagnostic Tool performs the following tests:

- Does the specified Web Service exist?
- Can the Predeployment Test Tool connect to the Web Service?
- Does the Web Service have an Application Catalog specified?
- Does the package have an Application Catalog specified?
- Displays the details of the target Application Catalog.
- Can the Web Service connect to the Application Catalog?
- Can the Web Service create and retrieve the test data?
- Can the results XML file be posted to the Web Service?

- Can the Web Service open the results XML file and validate the contents?

The Diagnostic Tool is embedded into test-ready packages, enabling you to test machines where the Predeployment Test tool is not installed, and is launched via the command line.

Running the Diagnostic Tool

When running in diagnostic mode, the test-ready package runs just as it does when not running in diagnostic mode. The difference is that when the results are posted, additional diagnostic information is displayed to assist in troubleshooting.

To have a test-ready package launch the Diagnostic Tool, set the PD_RUN_DIAGNOSTICS property. This property can be set:

- in the property table, or
- on the command line when launching the instrumented package.

```
msiexec.exe /i pkgname_analysis2.msi PD_RUN_DIAGNOSTICS=1
```



Task:

To run the Diagnostic Tool during test-ready package installation:

1. Open the Command Prompt and enter the following command to launch installation of the test-ready package:

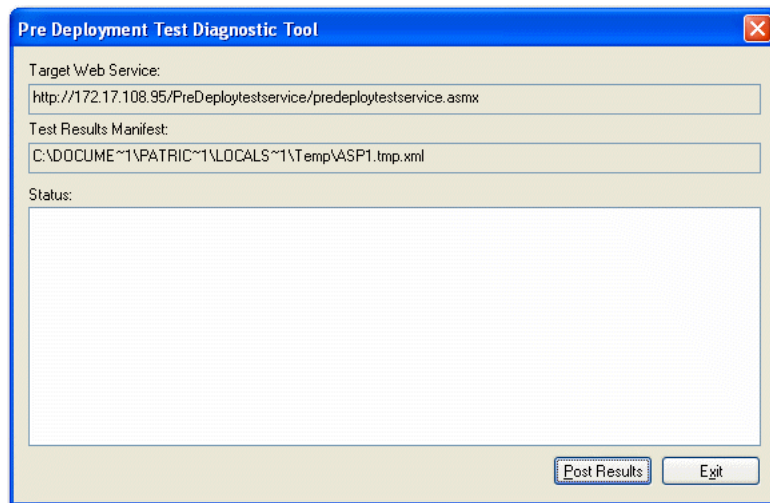
```
msiexec.exe /i pkgname_analysis.msi PD_RUN_DIAGNOSTICS=1
```

If you want to send the results to a log file in the same directory as the test-ready package, add the following parameter:

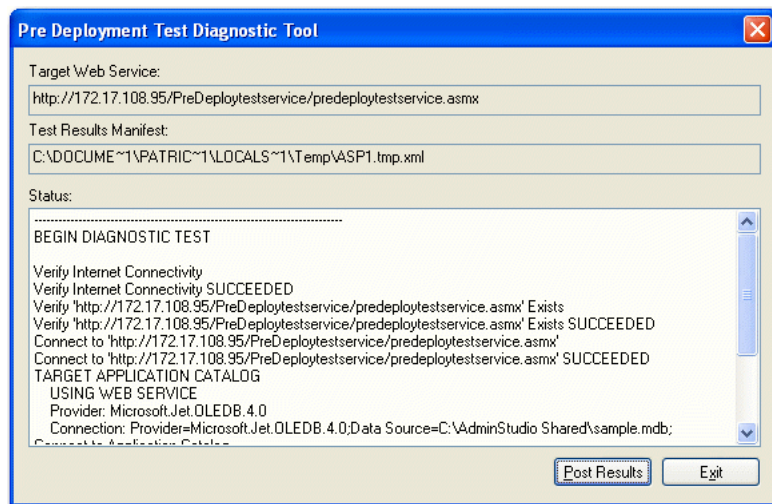
```
/!*v log.log
```

where *log.log* is the name of the log file.

The package installation will begin. When the installation is almost complete, the Diagnostic Tool interface opens:



2. Click **Post Results**. The results of the diagnostic tests are then listed in the **Status** window.



3. Review the results of the diagnostic tests. If you find an error message that could be resolved by making a minor adjustment to your system, such as adjusting your server settings, you could make that adjustment and click **Post Results** again. The Diagnostic Tool will run the diagnostic tests again (without having to launch the test-ready package installation again) and generate new results. You can then check to see if your adjustment fixed the problem.
4. When you have finished reviewing the results of the diagnostic tests, click **Exit**.

The installation of the test-ready package now finishes, and the results of the diagnostic tests are entered in the Predeployment Test Web Service Application Catalog, and, if you specified the log option, into the log file.



Note • If an instrumented package is run in silent mode and PD_RUN_DIAGNOSTICS is set, the Diagnostic Tool interface will not be displayed, but it will still run the diagnostic tests and write the results to the log file.

Viewing Predeployment Test Results

Use the **View Predeployment Test Results** link on the **Predeployment Test Start Page** to view the Predeployment Test Results.



Task: *To generate Predeployment Test results:*

1. Launch the Predeployment Test tool. The **Predeployment Test Start Page** opens.
2. Select **View Predeployment Test Results**. The [Predeployment Test Results By Job Report](#) opens in a Web browser, listing all of the test-ready packages that you have installed. All of the other reports can be accessed from this report.

The following table lists all of the available reports and how to access them:

Report Name	Description	How To View
Predeployment Test Results By Job Report	Lists all of the “test-ready” package jobs that have been run. Each time a test-ready package is installed on a machine in your network, test results are added to its corresponding job on the Predeployment Test Report Web site.	On the Predeployment Test Start Page , click View Predeployment Test Results .
Job Summary Report	Lists the total number of times the test-ready package was run, and the number of times it Passed , Failed , and was Incomplete .	On the Predeployment Test Results By Job Report , click the Job Name link.
Test Result Summary Report	Lists the all the computers that this job was run on (All Results), or all the computers that had either: Passed , Failed , or Incomplete results.	On the Job Summary Report , click the All Results , Passed , Failed , or Incomplete link. If the total number of tests in one of the result type categories is 0, the result type category name is not a link.
Job Details Report	Lists all of the deployment and conflict tests that were run on this machine for this job.	On a Test Result Summary Report , click on a Computer Name link.
Test Details Report	Lists detailed information on errors that were generated during a Deployment Test.	Click on a Test Name link in the Deployment Test Results list on the Job Details Report . (If no errors were found for that Test, the Test Name will not be a link.)
File/Registry Conflicts Report	Lists the results of the File Conflict checks (ACE07, ACE23) and the Registry Conflict checks (ACE10, ACE24).	On the Job Details Report , click the File Conflicts or Registry Conflicts link in the Conflict Test Results list.
Predeployment Test Results By Machine Report	Lists all of the machines in the network where “test-ready” packages have been installed.	On the Predeployment Test Results By Job Report , click the Click here to see results by Machine link.

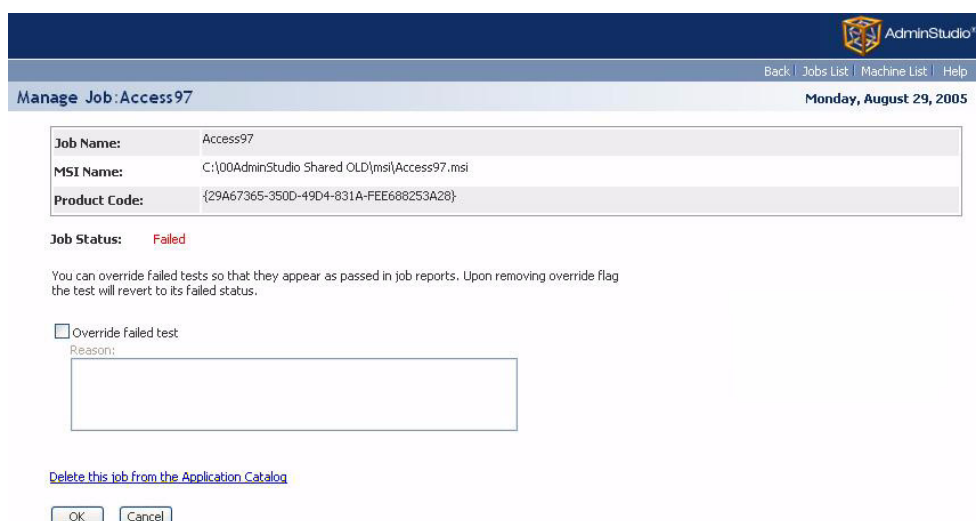
Report Name	Description	How To View
Machine Result Summary Report	Lists all of the test-ready package jobs run on a specific machine.	On the Predeployment Test Results By Machine Report , click on a Machine Name link.

Managing Job Data

After a Predeployment report is created, you can use the **Manage Job** feature to:

- Set a job with a **Failed** status to a **Passed** status, and enter a comment to explain why the status was changed.
- Delete the job.

If you click the **Manage Job** link on a **Job Summary Report** or **Job Details Report**, the **Manage Job Page** opens.



AdminStudio®

Back | Jobs List | Machine List | Help

Manage Job: Access97 Monday, August 29, 2005

Job Name: Access97

MSI Name: C:\00AdminStudio Shared OLD\msi\Access97.msi

Product Code: {29A67365-350D-49D4-831A-FEE688253A28}

Job Status: Failed

You can override failed tests so that they appear as passed in job reports. Upon removing override flag the test will revert to its failed status.

☐ Override failed test

Reason:

[Delete this job from the Application Catalog](#)

OK Cancel

On the [Manage Job Page](#), you can change the **Job Status** of a job from **Failed** to **Passed** or delete the job from the Application Catalog.

Changing the Status of a Job from Failed to Passed



Task: *To change the Job Status of a job:*

1. Open the **Job Summary Report** or **Job Details Report** for a **Failed** job and click **Manage Job**. The **Manage Job Page** opens.
2. Select the **Override failed test** check box in the **Job Status** area. The job's status is then changed to **Failed (Override to Passed)**.

Job Status: Failed (Override To Passed)

You can override failed tests so that they appear as passed in job reports. Upon removing override flag the test will revert to its failed status.

☒ Override failed test
Reason:
8/29/2005 2:10:34 PM

[Delete this job from the Application Catalog](#)

3. Enter a comment in the **Reason** text box to document why you changed this job's status and click **Save Changes**. The **Job Status** for this job is now listed as **Passed** in the Predeployment Test Results.

Marking a **Failed** job as being **Passed** does not modify any of the underlying test detail failure information that would normally be included in the **Job Details Report** or the **Test Details Report**. The original test results are always preserved for future reference, even though the predeployment test summary can be modified to appear as if the deployment tests succeeded.

Deleting a Job



Task: *To delete a job:*

1. Open the [Job Summary Report](#) or [Job Details Report](#) for a job and click **Manage Job**. The [Manage Job Page](#) opens.
2. Click **Delete this job from the Application Catalog**.

Managing a Job Globally or in the Context of a Machine

This **Manage Job** page can be used to manage a job globally or to manage the job only in the context of a given machine. When the **Manage Job** page is launched from a machine-specific report page, the name of the hyperlink will change to **Delete the job data for this machine** and the check box label will change to **Mark test as Passed for this machine**.

Predeployment Test Reference

Predeployment Test Reference includes the same topics displayed when you click a help button from a Wizard or dialog box in the Predeployment Test interface. The following reference topics are available:

- [Wizards and Dialog Boxes](#)
- [Technical Information](#)
- [Predeployment Test Results](#)
- [ConflictSolver ACE Reference](#)

Wizards and Dialog Boxes

- [Predeployment Test Start Page](#)
- [Predeployment Test Preparation Wizard](#)
- [Options Dialog Box](#)
- [Manage Job Page](#)
- [Configure Application Catalog for Predeployment Web Service Page](#)

Predeployment Test Start Page

The Predeployment Test tool allows system administrators to determine if a Windows Installer .msi package will succeed or fail when it is installed in production. AdminStudio will find issues with disk space, install conditions, and will find conflicts with .msi packages already installed on the system.

The Predeployment Test Start Page is launched by selecting the Predeployment Test icon on the AdminStudio Start Page or in the Tools Gallery. You can also launch it from the **Test Page** of the Process Assistant.

Predeployment testing is a three-step process:

Table 20-5 • Predeployment Test Process

#	Step	Description
1	Create “Test-Ready” Package	Select this option to launch the Predeployment Test Preparation Wizard to prepare a package for predeployment testing. See Creating a Test-Ready Windows Installer Package for more information.
2	Distribute “Test-Ready” Package	Predeployment testing does not occur until the test-ready package is distributed. Select the Launch Distribution Wizard option to distribute a test-ready .msi package. Or, you can distribute the test-ready package using another distribution method. See Distributing a Test-Ready Windows Installer Package .
3	View Test Results	Select this option to view the Predeployment Test Results. See Viewing Predeployment Test Results for more information.

Predeployment Test Preparation Wizard

The AdminStudio Predeployment Test feature helps to determine if a Windows Installer .msi package will succeed or fail when it is deployed in production. This wizard is the first step in the Predeployment analysis process.

The Predeployment Test Preparation Wizard consists of the following panels:

- [Welcome Panel](#)
- [Windows Installer File Selection Panel](#)
- [Web Service URL Panel](#)
- [Select Application Catalog Panel](#)
- [Deployment & Conflict Test Selection Panel](#)
- [Review Panel](#)
- [Progress Panel](#)
- [Summary Panel](#)

Welcome Panel

The first panel of the Predeployment Test Preparation Wizard welcomes you to the Wizard. Click **Next** to proceed to the [Windows Installer File Selection Panel](#).

This panel, and others in the Wizard, have four buttons located at the bottom of the Wizard. Depending on where you are in the Wizard, certain buttons may be disabled. The buttons are:

Table 20-6 • Wizard Buttons

Button	Description
Next	Advances you to the next panel in the Wizard.
Back	Moves you to the previous panel in the Wizard.
Cancel	Terminates the Wizard.
Help	Brings up help about the specific panel.

Windows Installer File Selection Panel

On this panel you specify the location of the original .msi package that is being prepared for deployment, and the name and location of the test-ready .msi package that you want to be generated at the end of this wizard.

Table 20-7 • Windows Installer File Selection Panel Options

Option	Description
Source .MSI Package	<p>Select the Windows Installer .msi package that you want to test with the Predeployment Test tool.</p> <p>In the Source .MSI Package field, select the Windows Installer .msi package that you want to test with the Predeployment Test tool—the .msi package that is being prepared for deployment.</p> <p>If you launched the Predeployment Test Preparation Wizard from the Application Manager by selecting a package and selecting Perform Pre-Deployment Testing from the context menu, the name in the Source .MSI Package field is already entered. The ability to edit this entry depends upon whether the package you are distributing is managed by the Software Repository:</p> <ul style="list-style-type: none"> • Not in the Software Repository—The full name and path of the Windows Installer file is displayed, and you can edit this entry or click Browse and select a different package. • In the Software Repository—Only the name of the Windows Installer file is displayed (not the full path) and this entry cannot be edited or changed.
Target .MSI Package	<p>Specify the name and path of the “test-ready” .msi package that this wizard will create. This field is automatically populated with the name of the package specified in the Source .MSI Package field with _analysis appended to the end of the name. For example, if WinZip.msi is selected as the Source .MSI Package, the Target .MSI Package will automatically be named WinZip_Analysis.msi. You can edit the name and the location of this file, if desired.</p> <p>If you launched the Predeployment Test Preparation Wizard from the Application Manager by selecting a package and selecting Perform Pre-Deployment Testing from the context menu, and the package is not in the Software Repository, the full name and path of the Target .MSI Package is displayed, but if the package is in the Software Repository, only the name of the Target .MSI Package is displayed (not the full path).</p>

Web Service URL Panel

On the Web Service URL Panel, enter the Web Service address where the Predeployment Test Web Service application is installed. Predeployment Test analysis test results will be posted to this Web Service.

When Predeployment Test is installed, the default location of the Predeployment Test Web Service is:

`http://IPADDRESS/PreDeploytestservice/predeploytestservice.asmx`

where *IPADDRESS* is the virtual directory address of the Predeployment Test Web Service application.

Table 20-8 • Web Service URL Panel Options

Option	Description
Predeployment Test Web Service URL	<p>Enter the HTTP Web Service URL where you want to post the Predeployment Test results. This field automatically displays the Web Service URL that was entered the last time the wizard was run.</p> <p>By default, this field is set to:</p> <p><code>http://adminstudioserver/PreDeployTestService/PreDeployTestService.asmx</code></p>
Job Identifier	<p>Enter a title to identify the Predeployment Test results in the analysis reports. By default, this field is populated with the package name.</p>



Select Application Catalog Panel

On this Panel, select the Application Catalog that will store the test results that are posted to the Predeployment Test Web Service.

Table 20-9 • Select Application Catalog Panel Options

Option	Description
Send test results to the Application Catalog registered with the web service	<p>Select this option to send the Predeployment Test results to the Application Catalog registered with the Web Service. That Application Catalog's Server and Catalog names are listed.</p> <p>If the Web Service selected on the Web Service URL Panel has been configured with an Application Catalog, this radio button will be selected by default the first time the Predeployment Test Preparation Wizard is run. On subsequent runs, this panel will default to the data entered in the previous run of the Wizard.</p>

Table 20-9 • Select Application Catalog Panel Options (cont.)

Option	Description
Send test results to an alternate Application Catalog	<p>Click Browse to open the Select Application Catalog Dialog Box and select an Application Catalog to receive the test results.</p> <p>If the Web Service selected on the Web Service URL Panel has not been configured with an Application Catalog, this radio button will be selected by default the first time the Predeployment Test Preparation Wizard is run, and the Application Catalog that AdminStudio is currently connected to will be selected. On subsequent runs, this panel will default to the data entered in the previous run of the Wizard.</p> <p>Registering is not required, but can significantly streamline the process of creating test-ready packages. In addition, when you select an “alternate” package, if you ever want to change the Application Catalog, you would have to rebuild all of your test-ready packages. With registered catalogs you just change the Application Catalog registered with the test service and then all of the new test runs will be committed to the newly registered catalog. It would be expected that the most common uses of the alternate package setting would be for people evaluating the tool and for people who need to occasionally target alternate Application Catalogs.</p>  <p>Note • If you select a Microsoft Access database, you must enter a path relative to the machine running the test service virtual directory.</p>  <p>Note • You are permitted to specify one Application Catalog for the Web Service and specify a different Application Catalog in the instrumented package. In this instance, the catalog information specified in the instrumented package takes precedence.</p>

Deployment & Conflict Test Selection Panel

On the **Deployment & Conflict Test Selection Panel**, select one or more **Deployment** and **Conflict** tests that you want to execute on the deployment machines when the test-ready package is deployed.

The following **Deployment** tests are available:

Table 20-10 • Deployment Tests

Test	Description
System Search	Locates a particular file, folder, registry key, .ini file value, or .xml file value on a target system prior to installation. Essentially, this feature lets you perform application, version, and configuration data searches.
Launch Conditions	Evaluates a set of conditions that must return True if the installation is to continue. If any condition fails, the user is presented with an error message and the installation ends.

Table 20-10 • Deployment Tests (cont.)

Test	Description
Directory Creation Permissions	Checks to determine if the necessary directory creation permissions exist for product installation.
File Read Permissions	Checks to determine if the necessary file read permissions exist for product installation.
Disk Space	Checks to determine if there is enough disk space for successful product installation.
Registry Permissions	Checks to determine if the necessary Registry permissions exist for product installation.
.NET Framework	Checks to see if the .NET Framework was installed correctly.
File Association	Checks to see if all file extensions have been installed and associated correctly. This involves launching a file with this extension, and determining if the correct application was used.

The following **Conflict** tests are available:

Table 20-11 • Conflict Tests

Test	Description
File	File Conflict Checks (ACE07 , ACE23).
Registry	Registry Checks (ACE10 , ACE24).

Selecting a folder automatically selects all tests in that folder, and selecting the **Predeployment Tests** folder automatically selects all **Deployment** and **Conflict** tests in all folders.



Note • The **Deployment & Conflict Test Selection Panel** automatically displays the selections that were made during the previous run of this wizard. These selections are saved in an XML file, *PDTests.xml*, in the **Support** subfolder of the AdminStudio installation directory.



Note • If you select to include File or Registry conflict tests, it will take a longer time to complete the analysis during deployment of a test-ready package.

Review Panel

On this panel you can review the settings you specified in this wizard, and then initiate the testing.

Click **Start** to initiate the generation of the test-ready .msi package and proceed to the **Progress Panel**. To edit a setting before initiating testing, click **Back** to return to previous panels.



Note • If a test-ready .msi file with same name as the package you are generating already exists, it will be overwritten.

Progress Panel

The Progress Panel indicates the progress of the testing. When testing is complete, the **Next** button is activated. Click **Next** to proceed to the **Summary Panel**.

Summary Panel

The Summary Panel lets you know that the test-ready .msi package was successfully generated, and lists a summary of the package settings.

Click **Finish** to exit the Wizard.

Or, to automatically launch the **Distribution Wizard** so that you can distribute this test-ready .msi package, select the **Distribute the test-ready Windows Installer .msi package** now option and click **Finish**. The Distribution Wizard will be launched with the generated test-ready .msi package automatically selected as the distribution source.

Options Dialog Box

Enter information on the Options dialog box to configure the Predeployment Test Results Web Site and the Predeployment Test Web Service. To open the **Options** dialog box, open the **Predeployment Test Start Page** and click **Options** on the **Tools** menu.

Table 20-12 • Predeployment Options Dialog Box Options

Option	Description
Predeployment Test Results Web Site	<p>Configure the Predeployment Test Results Web Site, the Web location where predeployment test results are displayed in Report Format. See Predeployment Test Results for report examples.</p> <p>Specify the following:</p> <ul style="list-style-type: none">• URL—Enter the URL of the Predeployment Test Results Web site.• Configure Application Catalog—Click to open the Assigning ASP.NET Permissions to Files or Folders to configure a connection to an Application Catalog to store the Predeployment Test results.

Table 20-12 • Predeployment Options Dialog Box Options (cont.)

Option	Description
Predeployment Test Web Service	<p>Configure the Predeployment Test Web Service, the application that receives predeployment test result data.</p> <p>Specify the following:</p> <ul style="list-style-type: none">• URL—Enter the URL of the Predeployment Test Web Service.• Configure Application Catalog—Click to open the Assigning ASP.NET Permissions to Files or Folders to configure a connection to an Application Catalog that will store the test result data generated when test-ready packages are installed.

Configure Application Catalog for Predeployment Web Service Page

In Predeployment Test, you can connect to an SQL Server Application Catalog. The procedure for opening an Application Catalog is the same as the procedure used in the AdminStudio interface, Application Manager, and ConflictSolver. See [Connecting to an Existing Application Catalog](#).

Technical Information

This section includes the following topics:

- [Predeployment Custom Properties](#)
- [Predeployment Test Identifiers](#)
- [Predeployment Test Results XML File](#)
- [Changes Made During Test-Ready Package Creation](#)

Predeployment Custom Properties

The **Predeployment Test Preparation Wizard** adds some custom properties to the test-ready MSI package. These properties, which are used by the custom actions, can be passed as parameters to msixec.exe to override their values in the .msi file.

Table 20-13 • Predeployment Custom Properties

Property	Description
PDTESTSTORUN	Comma separated list of deployment and conflict test identifiers to run. See Predeployment Test Identifiers for a list of these tests. If this property value is an asterisk (*), it indicates that all the tests will be run. This list populates the list shown on the Deployment & Conflict Test Selection Panel . The selections made on this panel are saved in an XML file (PDTests.xml in the Support subfolder of the AdminStudio program folder) so that the same test selections are displayed the next time the wizard is run.
PDJOBID	The descriptive test identifier.
PDPKGSRC	The original Windows Installer .msi package file.
PDWEBSERVICEURL	Web Service URL where the Predeployment Test results will be posted.

Predeployment Test Identifiers

The following terms identify the tests that are run during Predeployment testing:

AppSearch
LaunchCondition
CreateFolder
FileRead
FileWrite
DiskSpace
Registry
DotNetFramework
FileAssociation
FileConflicts
RegConflicts

A comma separated list of the above identifiers is saved as a property, PDTestsToRun, in the test-ready .msi package. This property value is then used by the custom actions to determine the tests to run during the analysis. The list of tests in the PDTestsToRun property populates the list shown on the [Deployment & Conflict Test Selection Panel](#).

Predeployment Test Results XML File

Test results are sent to a .NET Web service and are saved into database tables in the Application Catalog. The Predeployment Test tool accesses the saved test results in the database tables and displays them in a Web browser. This is an example of an XML file containing those test results.

```
<div class="dtd-fragment">
<pre>
<Predeployment version="1.0" ComputerName="SCH01SINGHDXP"
  IP="172.17.100.22" OS="Windows Server 2003 version 5.2
  (Build 3790)" Language="English" UserName="JohnSmith"
  JobID="CorelDraw" RunStartDate="6/15/2004 18:19"
  xmlns="http://www.installshield.com/web services/">
<Original>
  <PackageCode>{FAD3134F-5474-4778-9E34-204662673F6D}</PackageCode>
  <ProductCode>{5A92C452-89F2-435C-8C13-26EE70F51059}</ProductCode>
  <Name>Corel Draw</Name>
  <Version>9.5</Version>
  <Language>1033</Language>
  <Source>\\server\SharedUsers\JohnSmith\_Performance\MSIs\
    Corel Draw.msi</Source>
</Original>
<Instrumentation NewPackageCode="{FAD3134F-5474-4778-9E34-
  204662673F6D}" NewProductCode="
  {59293687-ABCF-4e96-9CF5-5F3A08E2166A}" />
<TestResults>
  <Deployment>
    <Test ID="AppSearch">
      <Result Total="0" Success="0" Failed="0" />
    </Test>
    <Test ID="LaunchCondition">
      <Result Total="0" Success="0" Failed="0" />
    </Test>
    <Test ID="CreateFolder">
      <Message Key="INSTALLDIR" Folder="C:\Program Files\Corel\
        Graphics9\" Result="0" />
      <Message Key="Directory26" Folder="C:\Program Files\
        Corel\Graphics9\Workspace\Corel Photo-Paint9
        _BootDefault\" Result="0" />
      <Message Key="WindowsFolder" Folder="C:\WINDOWS\" Result="0" />
      <Message Key="MESHWARD" Folder="C:\Program Files\Corel\
        Graphics9\Custom\MeshWarp\" Result="0" />
      <Message Key="TEMPLATE" Folder="C:\Program Files\Corel\
        Graphics9\Draw\Template\" Result="0" />
      <Message Key="PAPERTYPES" Folder="C:\Program Files\Corel\
        Graphics9\Papertypes\" Result="0" />
      <Message Key="SYMBOLS" Folder="C:\Program Files\Corel\
        Graphics9\Symbols\" Result="0" />
      <Message Key="BOOTDEFAULT" Folder="C:\Program Files\Corel\
        Graphics9\Workspace\CorelDRAW9\_BootDefault\" Result="0" />
      <Message Key="PROFILES" Folder="C:\WINDOWS\Profiles\"
        Result="0" />
      <Message Key="INSTALLDIR.A67717E9_4C97_4D36_947C_FC3157B0BD67"
        Folder="C:\WINDOWS\system32\" Result="0" />
      <Message Key="DIR51" Folder="C:\Documents and Settings\
```

```

    JohnSmith\Favorites\core1 on the web\" Result="0" />
<Message Key="DIR52" Folder="C:\Program Files\Core1\
  Graphics9\color\" Result="0" />
<Message Key="DIR53" Folder="C:\Program Files\Core1\
  Graphics9\config\" Result="0" />
<Message Key="DIR54" Folder="C:\Program Files\Core1\
  Graphics9\custom\" Result="0" />
<Message Key="DIR55" Folder="C:\Program Files\Core1\
  Graphics9\custom\brushes\" Result="0" />
<Message Key="DIR56" Folder="C:\Program Files\Core1\
  Graphics9\custom\bumpmap\" Result="0" />
<Message Key="DIR57" Folder="C:\Program Files\Core1\
  Graphics9\custom\canvas\" Result="0" />
<Message Key="DIR58" Folder="C:\Program Files\Core1\
  Graphics9\custom\displace\" Result="0" />
<Message Key="DIR59" Folder="C:\Program Files\Core1\
  Graphics9\custom\duotone\" Result="0" />
<Message Key="DIR60" Folder="C:\Program Files\Core1\
  Graphics9\custom\frames\" Result="0" />
<Message Key="DIR61" Folder="C:\Program Files\Core1\
  Graphics9\custom\layouts\" Result="0" />
<Message Key="DIR62" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\" Result="0" />
<Message Key="DIR64" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\cmyk\misc\" Result="0" />
<Message Key="DIR65" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\cmyk\nature\" Result="0" />
<Message Key="DIR66" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\cmyk\people\" Result="0" />
<Message Key="DIR67" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\cmyk\things\" Result="0" />
<Message Key="DIR69" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\rgb\misc\" Result="0" />
<Message Key="DIR70" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\rgb\nature\" Result="0" />
<Message Key="DIR71" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\rgb\people\" Result="0" />
<Message Key="DIR72" Folder="C:\Program Files\Core1\
  Graphics9\custom\palettes\rgb\things\" Result="0" />
<Message Key="DIR73" Folder="C:\Program Files\Core1\
  Graphics9\custom\patterns\" Result="0" />
<Message Key="DIR74" Folder="C:\Program Files\Core1\
  Graphics9\custom\shearmap\" Result="0" />
<Message Key="DIR75" Folder="C:\Program Files\Core1\
  Graphics9\custom\tiles\" Result="0" />
<Message Key="DIR76" Folder="C:\Program Files\Core1\
  Graphics9\custom\tonecrve\" Result="0" />
<Message Key="DIR77" Folder="C:\Program Files\Core1\
  Graphics9\custom\userdef\" Result="0" />
<Message Key="DIR78" Folder="C:\Program Files\Core1\
  Graphics9\draw\" Result="0" />
<Message Key="DIR79" Folder="C:\Program Files\Core1\
  Graphics9\draw\custommediastrokes\" Result="0" />
<Message Key="DIR82" Folder="C:\Program Files\Core1\
  Graphics9\draw\fill_out\fills\fountain\" Result="0" />
<Message Key="DIR83" Folder="C:\Program Files\Core1\

```

```
Graphics9\draw\fill_out\fills\fountain\cylinder\"
Result="0" />
<Message Key="DIR84" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\fountain\radial\"
Result="0" />
<Message Key="DIR86" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\pattern\2color\" Result="0" />
<Message Key="DIR87" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\pattern\bitmap\" Result="0" />
<Message Key="DIR88" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\pattern\color\" Result="0" />
<Message Key="DIR89" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\texture\" Result="0" />
<Message Key="DIR90" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\texture\biology\"
Result="0" />
<Message Key="DIR91" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\texture\elements\"
Result="0" />
<Message Key="DIR92" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\texture\material\"
Result="0" />
<Message Key="DIR93" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\fills\texture\rocks\"
Result="0" />
<Message Key="DIR94" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\outlines\" Result="0" />
<Message Key="DIR95" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\outlines\arrows\" Result="0" />
<Message Key="DIR96" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\outlines\dot_dash\" Result="0" />
<Message Key="DIR97" Folder="C:\Program Files\Core1\
Graphics9\draw\fill_out\outlines\width\" Result="0" />
<Message Key="DIR98" Folder="C:\Program Files\Core1\
Graphics9\draw\find\" Result="0" />
<Message Key="DIR99" Folder="C:\Program Files\Core1\
Graphics9\draw\net_fav\" Result="0" />
<Message Key="DIR100" Folder="C:\Program Files\Core1\
Graphics9\draw\net_fav\links\" Result="0" />
<Message Key="DIR101" Folder="C:\Program Files\Core1\
Graphics9\draw\samples\" Result="0" />
<Message Key="DIR103" Folder="C:\Program Files\Core1\
Graphics9\draw\scripts\presets\" Result="0" />
<Message Key="DIR104" Folder="C:\Program Files\Core1\
Graphics9\draw\scripts\scripts\" Result="0" />
<Message Key="DIR105" Folder="C:\Program Files\Core1\
Graphics9\filters\" Result="0" />
<Message Key="DIR106" Folder="C:\Program Files\Core1\
Graphics9\photopnt\" Result="0" />
<Message Key="DIR107" Folder="C:\Program Files\Core1\
Graphics9\photopnt\brushtxr\" Result="0" />
<Message Key="DIR108" Folder="C:\Program Files\Core1\
Graphics9\photopnt\imglists\" Result="0" />
<Message Key="DIR109" Folder="C:\Program Files\Core1\
Graphics9\photopnt\net_fav\" Result="0" />
<Message Key="DIR110" Folder="C:\Program Files\Core1\
```

```

Graphics9\photopnt\net_fav\links\" Result="0" />
<Message Key="DIR111" Folder="C:\Program Files\Core1\
Graphics9\photopnt\paths\" Result="0" />
<Message Key="DIR112" Folder="C:\Program Files\Core1\
Graphics9\photopnt\samples\" Result="0" />
<Message Key="DIR114" Folder="C:\Program Files\Core1\
Graphics9\photopnt\scripts\effects\" Result="0" />
<Message Key="DIR115" Folder="C:\Program Files\Core1\
Graphics9\photopnt\scripts\scripts\" Result="0" />
<Message Key="DIR116" Folder="C:\Program Files\Core1\
Graphics9\photopnt\tables\" Result="0" />
<Message Key="DIR117" Folder="C:\Program Files\Core1\
Graphics9\plugins\" Result="0" />
<Message Key="DIR118" Folder="C:\Program Files\Core1\
Graphics9\plugins\digimarc\" Result="0" />
<Message Key="DIR119" Folder="C:\Program Files\Core1
Graphics9\plugins\squizz\" Result="0" />
<Message Key="DIR120" Folder="C:\Program Files\Core1\
Graphics9\programs\" Result="0" />
<Message Key="DIR121" Folder="C:\Program Files\Core1\
Graphics9\programs\data\" Result="0" />
<Message Key="DIR122" Folder="C:\Program Files\Core1\
Graphics9\register\" Result="0" />
<Message Key="DIR123" Folder="C:\Program Files\Core1\
Graphics9\scripts\" Result="0" />
<Message Key="DIR125" Folder="C:\Program Files\Core1\
Graphics9\shared\help\" Result="0" />
<Message Key="DIR126" Folder="C:\Program Files\Core1\
Graphics9\trace\" Result="0" />
<Message Key="DIR127" Folder="C:\Program Files\Core1\
Graphics9\trace\samples\" Result="0" />
<Message Key="DIR130" Folder="C:\Program Files\Core1\
Graphics9\tutors\draw\htmldocs\" Result="0" />
<Message Key="DIR131" Folder="C:\Program Files\Core1\
Graphics9\tutors\draw\htmldocs\htmlpics\" Result="0" />
<Message Key="DIR133" Folder="C:\Program Files\Core1\
Graphics9\tutors\paint\htmldocs\" Result="0" />
<Message Key="DIR134" Folder="C:\Program Files\Core1\
Graphics9\tutors\paint\htmldocs\htmlpics\" Result="0" />
<Message Key="DIR136" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1 photo-paint9\" Result="0" />
<Message Key="DIR137" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1 photo-paint9\adobe(r)
photoshop(r)\" Result="0" />
<Message Key="DIR138" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1 photo-paint9\core1
photo-paint 8\" Result="0" />
<Message Key="DIR139" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1draw9\" Result="0" />
<Message Key="DIR140" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1draw9\adobe(r) illustrator(r)\"
Result="0" />
<Message Key="DIR141" Folder="C:\Program Files\Core1\
Graphics9\workspace\core1draw9\core1draw8\" Result="0" />
<Message Key="DIR145" Folder="C:\Program Files\core1\
shared\writing tools\9.0\" Result="0" />

```



```

<Message Key="DIR146" Folder="C:\WINDOWS\corel\" Result="0" />
<Message Key="DIR148" Folder="C:\WINDOWS\corel\
  coreldraw 9\9\" Result="0" />
<Message Key="DIR149" Folder="C:\Program Files\Corel\
  Graphics9\Filters\convert\" Result="0" />
<Message Key="SystemFolder" Folder="C:\WINDOWS\system32\"
  Result="0" />
<Message Key="DIR29.5B60FF9E_851D_11D4_A752_00B0D0428C0C"
  Folder="Y:\Program Files\Common Files\Microsoft
  Shared\DAO\" Result="0" />
<Message Key="INSTALLDIR.46C3930E_0E70_11D5_BF08_0010A4EF69FF"
  Folder="C:\Program Files\Borland\Common Files\Default\"
  Result="0" />
<Message Key="Dir388" Folder="C:\Documents and Settings\
  JohnSmith\Start Menu\Programs\CorelDRAW 9\
  Graphics Utilities\" Result="0" />
<Message Key="Dir387" Folder="C:\Documents and Settings\
  JohnSmith\Start Menu\Programs\CorelDRAW 9\" Result="0" />
<Message Key="Dir389" Folder="C:\Documents and Settings\
  JohnSmith\Start Menu\Programs\CorelDRAW 9\
  Productivity Tools\" Result="0" />
<Message Key="Dir390" Folder="C:\Documents and Settings\
  JohnSmith\Start Menu\Programs\CorelDRAW 9\
  Setup and Notes\" Result="0" />
<Message Key="Directory16" Folder="C:\Program Files\Corel\
  Graphics9\Config\" Result="0" />
<Message Key="Directory17" Folder="C:\Program Files\Corel\
  Graphics9\Custom\Tiles\" Result="0" />
<Message Key="Directory5" Folder="C:\Program Files\Corel\
  Graphics9\Draw\" Result="0" />
<Message Key="Directory19" Folder="C:\Program Files\Corel\
  Graphics9\Filters\" Result="0" />
<Message Key="Directory20" Folder="C:\Program Files\Corel\
  Graphics9\Photopnt\" Result="0" />
<Message Key="Directory22" Folder="C:\Program Files\Corel\
  Graphics9\Programs\Data\" Result="0" />
<Message Key="Directory23" Folder="C:\Program Files\Corel\
  Graphics9\Register\" Result="0" />
<Message Key="Directory24" Folder="C:\Program Files\Corel\
  Graphics9\Trace\" Result="0" />
<Message Key="Directory27" Folder="C:\Program Files\Corel\
  Graphics9\Workspace\Corel Photo-Paint9\
  Adobe(R) Photoshop(R)\\" Result="0" />
<Message Key="TempFolder" Folder="C:\DOCUME~1\DARSHA~1\
  LOCALS~1\Temp\" Result="0" />
<Message Key="CommonFilesFolder" Folder=
  "Y:\Program Files\Common Files\" Result="0" />
<Message Key="StartupFolder" Folder="C:\Documents and Settings\
  JohnSmith\Start Menu\Programs\Startup\" Result="0" />
<Result Total="112" Success="112" Failed="0" />
</Test>
<Test ID="FileRead">
  <Result Total="3399" Success="3399" Failed="0" />
</Test>
<Test ID="FileWrite">
  <Result Total="3399" Success="3399" Failed="0" />

```

```
</Test>
<Test ID="Registry">
  <Result Total="802" Success="802" Failed="0" />
</Test>
<Test ID="DotNetFramework">
  <Message AssembliesCount="0" V10="0" V11="1" V12="0" V20="0"
    InstallRoot="C:\WINDOWS\Microsoft.NET\Framework\" />
  <Result Total="1" Success="1" Failed="0" />
</Test>
<Test ID="DiskSpace">
  <Result Total="1" Success="1" Failed="0" />
</Test>
<Test ID="FileAssociation">
  <Result Total="11" Success="11" Failed="0" />
</Test>
</Deployment>
<Conflict>
</Conflict>
</TestResults>
<Conclusion>
  <RunEndDate>6/15/2004 18:20</RunEndDate>
  <DeploymentTests Run="9" Success="9" Failed="0" />
  <ConflictTests Run="0" Success="0" Failed="0" />
</Conclusion>
</Predeployment>
</pre>
</div>
```

Changes Made During Test-Ready Package Creation

The Predeployment Test Preparation Wizard makes a copy of the selected source Windows Installer .msi package and makes the following changes to create a “test-ready” package. These changes are made to ensure that the test-ready package does not make any system changes when it is deployed.

- Add a Type 1 custom action for each test into the **CustomAction** table.
- Add or update the **EXECUTEMODE** property and set its value to None.
- Add or update the **REBOOT** property and set its value to ReallySuppress.
- Add or update the **ARPNO MODIFY** property and set its value to 1.
- Add or update the **ARPNO REMOVE** property and set its value to 1.
- Add or update the **ARPNO REPAIR** property and set its value to 1.
- Update the product code, and save the original product code in a different, new property.
- Update the product code in the Product table.
- Add other properties, such as Web service URL, test descriptive identifier, tests to run, etc. based on the data connected in the wizard.
- Remove the existing custom actions and their references from all the user interface and execute sequence tables.

- Update the product name.
- Delete all rows from **_Validation** table.

Predeployment Test Results

The following table lists all of the available Predeployment Test results and how to access them on the Predeployment Test Report Web site:

Table 20-14 • Predeployment Test Results

Report Name	Description	How To View
Predeployment Test Results By Job Report	Lists all of the “test-ready” package jobs that have been run. Each time a test-ready package is installed on a machine in your network, test results are added to its corresponding job on the Predeployment Test Report Web site.	On the Predeployment Test Start Page , click View Predeployment Test Results .
Job Summary Report	Lists the total number of times the test-ready package was run, and the number of times it Passed , Failed , and was Incomplete .	On the Predeployment Test Results By Job Report , click the Job Name link.
Test Result Summary Report	Lists the all the computers that this job was run on (All Results), or all the computers that had either: Passed , Failed , or Incomplete results.	On the Job Summary Report , click the All Results , Passed , Failed , or Incomplete link. If the total number of tests in one of the result type categories is 0, the result type category name is not a link.
Job Details Report	Lists all of the deployment and conflict tests that were run on this machine for this job.	On a Test Result Summary Report , click on a Computer Name link.
Test Details Report	Lists detailed information on errors that were generated during a Deployment Test.	Click on a Test Name link in the Deployment Test Results list on the Job Details Report . (If no errors were found for that Test, the Test Name will not be a link.)
File/Registry Conflicts Report	Lists the results of the File Conflict checks (ACE07, ACE23) and the Registry Conflict checks (ACE10, ACE24).	On the Job Details Report , click the File Conflicts or Registry Conflicts link in the Conflict Test Results list.

Table 20-14 • Predeployment Test Results (cont.)


Report Name	Description	How To View
Predeployment Test Results By Machine Report	Lists all of the machines in the network where “test-ready” packages have been installed.	On the Predeployment Test Results By Job Report , click the Click here to see results by Machine link.
Machine Result Summary Report	Lists all of the test-ready package jobs run on a specific machine.	On the Predeployment Test Results By Machine Report , click on a Machine Name link.

The following Reference topics are also included:

- [Connection to New Catalog Page](#)
- [Manage Job Page](#)
- [Assigning ASP.NET Permissions to Files or Folders](#)

Predeployment Test Results By Job Report

The Predeployment Test Results By Job report lists all of the “test-ready” package jobs that have been run. Each time a test-ready package is installed on a machine in your network, test results are added to its corresponding job on the Predeployment Test Report Web site. This report is the first page you see when opening the Predeployment Test Report Web site.


AdminStudio

Application Catalog : sch01thornleyxp\Development(howesqlstandalone)
Change Application Catalog | Help

Predeployment Test Results by Job

Monday, August 29, 2005

[Results by Job](#)

Job Name	Product Name	File Name	Product Version	No. of times run	Run Date
Access97	Access97_Analysis	Access97.msi	97	1	8/29/2005 1:31:00 PM
AspectArchitect	AspectArchitect_Analysis	AspectArchitect.msi	2.1	1	8/29/2005 1:10:00 PM
Brio Navigator	Brio Navigator_Analysis	Brio Navigator.msi	5.5	1	8/29/2005 1:29:00 PM
DemoPlayer	DemoPlayer_Analysis	DemoPlayer.msi	1.00.0000	1	8/29/2005 12:59:00 PM
1					

[Click here](#) to see results by Machine.

The following information is included:

Table 20-15 • Predeployment Test Results By Job Report Data


Item	Description
Job Name	Name of the test-ready package job. Corresponds to the Job Identifier that was entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard when this test-ready package was created. Click on this link to view the Job Summary Report for this job.

Table 20-15 • Predeployment Test Results By Job Report Data (cont.)

Item	Description
Product Name	Name of the test-ready .msi package associated with this job. (The .msi extension is not shown.)
File Name	Name of the original .msi package that was used to create the test-ready package.
Product Version	Version of the original .msi package.
No. of times run	Number of times this job has been run (this test-ready package has been installed) on computers in your network.
Run Date	Date and time that this job was last run.
Change Application Catalog	Click to access the Connection to New Catalog Page , where you can open a new Application Catalog while you are using the Predeployment Test Results Web site.
Click here to see results by machine	Click to view the Predeployment Test Results By Machine Report .

Job Summary Report

The Job Summary report lists the total number of times the test-ready package was run, and the number of times it Passed, Failed, and was Incomplete. To access this report, go to the [Predeployment Test Results By Job Report](#), and click the **Job Name** link.

 AdminStudio®	
Jobs List Machine List Help	
Job Summary: Access97	
Monday, August 29, 2005	
Job Name:	Access97
MSI Name:	C:\00AdminStudio Shared OLD\msi\Access97.msi
Product Code:	{29A67365-350D-49D4-831A-FEE688253A28}
First Run On:	8/29/2005 1:31:00 PM
Last Run On:	8/29/2005 1:33:00 PM
No. of Times run:	1
Results:	
All Results	1 (100%)
Passed	0 (0%)
Failed	1 (100%)
Incomplete	0 (0%)
Manage Job	

The following information is included:


Table 20-16 • Job Summary Report Data


Item	Description
Job Name	Name of the test-ready package job. Corresponds to the Job Identifier that was entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard when this test-ready package was created. Click on this link to view the Job Summary Report for this job.
MSI Name	Name and location of the test-ready .msi package associated with this job.
Product Code	Product GUID of the .msi package that was used to create the test-ready package.
First Run On & Last Run On	Date the test-ready package was run the first time and the most recent time.
No. of Times run	Number of times this job has been run (this test-ready package has been installed) on computers in your network.
Results	Four category totals are provided to summarize the test results: All Results, Passed, Failed, and Complete. Click on one of these links to view the Test Result Summary Report for the selected category. If there were no results in a category, the name of that category is not a link.
Manage Job	Click to open the Manage Job Page , where you can choose to override a failed test or delete this Job from the Application Catalog.
Jobs List	Click to access the Predeployment Test Results By Job Report .
Machine List	Click to access the Predeployment Test Results By Machine Report .

Test Result Summary Report

The Test Results Summary reports list all the computers that this test-ready package job was run on (for the All Results report), or all the computers where the test-ready package tests Passed, Failed, or had Incomplete results.

To access the Test Results Summary Reports, open the [Job Summary Report](#), and click the All Results, Passed, Failed, or Incomplete link. If the total number of tests in one of the result type categories is 0, the result type category name is not a link.

<div>  AdminStudio® </div>	
Back Jobs List Machine List Summary of Access97 Help	
Test Result Summary: All Results Monday, August 29, 2005	
Job Name:	Access97
MSI Name:	C:\00AdminStudio Shared OLD\msi\Access97.msi
Product Code:	{29A67365-350D-49D4-831A-FEE688253A28}
Total tests run on each machine:	11

Computer Name	Operating System	Start Time	Result	Deployment Tests Failed	Conflict Tests Failed
SCHOIHOWEPXP	Microsoft Windows XP version 5.1 Service Pack 2 (Build 2600)	8/29/2005 1:31:00 PM		1	2
1					

The following information is included:

Table 20-17 • Test Results Summary Report Data

Item	Description
Job Name	Name of the test-ready package job. Corresponds to the Job Identifier that was entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard when this test-ready package was created. Click on this link to view the Job Summary Report for this job.
MSI Name	Name and location of the test-ready .msi package associated with this job.
Product Code	Product GUID of the .msi package that was used to create the test-ready package.
Total tests run on each machine	The total number of Deployment and Conflict tests associated with this test-ready package.
Computer Name	Unique identifier of an individual computer in the network.
Operating System	Operating system of the machine.
Start Time	Time the installation of the test-ready package started on this machine.
Result	Icon indicating whether the test-ready package installation (and associated tests) Passed (✓), Failed (✗), or were Incomplete .
Deployment Tests Failed	Total number of deployment tests that failed during the installation of the test-ready package on this computer.
Conflict Tests Failed	Total number of conflict tests that failed during the installation of the test-ready package on this computer.
Jobs List	Click to access the Predeployment Test Results By Job Report .

Table 20-17 • Test Results Summary Report Data (cont.)

Item	Description
Machine List	Click to access the Predeployment Test Results By Machine Report .
Summary of [Test Name]	Click to view the Job Summary Report for this test-ready package job.

Job Details Report

The Job Details report lists all of the deployment and conflict tests that were run on this machine for this job. To access it, click the Computer Name link on a [Test Result Summary Report](#).

The screenshot displays the 'Job Details: Access97' report in AdminStudio. The report includes the following information:

- Job Name:** Access97
- MSI Name:** C:\00AdminStudio Shared OLD\msi\Access97.msi
- Product Code:** {29A67365-350D-49D4-831A-FEE688253A28}

Deployment Test Results:

Test Name	Message	Status
AppSearch	No results for AppSearch test.	✖
Launch Conditions	Checked 2 Launch Conditions. 2 passed & 0 failed.	✔
Folder Create - Security Access	Checked creation permissions for 19 folders. 19 passed and 0 failed.	✔
File - Read Permission	Checked Read security permission for 227 files. 227 passed and 0 failed.	✔
File - Write Permission	Checked Write security permission for 227 files. 227 passed and 0 failed.	✔
Registry - Write Permission	Checked Write security permission for 2448 Registry entries. 2448 passed and 0 failed.	✔
.Net Framework Install status	.NET Framework is installed.	✔
DiskSpace requirements	Checked the Disk Space requirements for this package.	✔
File Associations	Checked 30 File Associations against current File Associations on the machine. 2 passed and 28 failed.	✖

Conflict Test Results:

Test Name	Message	Status
File Conflicts	Checked for File conflicts with the MSI packages installed on the Target Machine. 27 conflicts found.	✖
Registry Conflicts	Checked for Registry conflicts with the MSI packages installed on the Target Machine. 496 conflicts found.	✖

[Manage Job](#)

The following information is included:

Table 20-18 • Job Details Report Data

Item	Description
Job Name	Name of the test-ready package job. Corresponds to the Job Identifier that was entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard when this test-ready package was created. Click on this link to view the Job Summary Report for this job.
MSI Name	Name and location of the test-ready .msi package associated with this job.
Product Code	Product GUID of the .msi package that was used to create the test-ready package.

Table 20-18 • Job Details Report Data (cont.)

Item	Description
Deployment and Conflict Test Results	<p>For each deployment and conflict test that was included with this test-ready package, the following information listed:</p> <ul style="list-style-type: none"> • Test Name—Name of the deployment or conflict test that was included with this test-ready package. • Message—Summary message generated when this deployment or conflict test was run on this computer indicating quantity of items (folders, files, Registry entries, etc.) checked, and how many of those items passed the test, and how many failed. • Status—Icon indicating the deployment or conflict test status: Passed (🟢), Failed (🔴), or Incomplete.
Manage Job	Click to open the Manage Job Page , where you can choose to override a failed test or delete this Job from the Application Catalog.
Jobs List	Click to access the Predeployment Test Results By Job Report .
Machine List	Click to access the Predeployment Test Results By Machine Report .

Test Details Report

The Test Details report lists detailed information on errors that were generated during a Deployment Test. To access it, click on a Test Name link in the **Deployment Test Results** list on the [Job Details Report](#). (If no errors were found for that Test, the Test Name will not be a link.)

AdminStudio®		
Back Jobs List Machine List Summary of Access97 Help		
Test Details: File Association		Monday, August 29, 2005
Job Name:	Access97	
Computer Name:	SCH01HOWEPXP	
File Association Key	File Association Verb	Test Status
mad	open	Warning
maf	open	Warning
maf	datasheet	Warning
maf	design	Warning
maf	preview	Warning
maf	print	Warning
maf	printto	Warning
mam	open	Warning
mam	design	Warning
maq	open	Warning
maq	design	Warning
maq	preview	Warning
maq	print	Warning
maq	printto	Warning
mar	open	Warning
1 2		


The following information is included:

Table 20-19 • Test Details Report Data

Item	Description
Job Name	Name of the test-ready package job. Corresponds to the Job Identifier that was entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard when this test-ready package was created. Click on this link to view the Job Summary Report for this job.
Computer Name	Unique identifier of an individual computer in the network.
[Conflicting Item in Product]	The item in a product installed on the computer that the test-ready package was installed on that conflicted with something in the test-ready package. The information displayed in this report varies depending upon the type of test that you are viewing.
Test Status	Icon indicating whether the item generated a warning, failed to install: (❌), successfully installed: (✅), or was Incomplete.
Jobs List	Click to access the Predeployment Test Results By Job Report .
Machine List	Click to access the Predeployment Test Results By Machine Report .
Summary of [Job Name]	Click to access the Job Summary Report for this test-ready package job.

File/Registry Conflicts Report

The **File / Registry Conflicts Report** lists File or Registry conflicts that were detected against the selected package. To open this report, click the **File Conflicts** or **Registry Conflicts** link in the **Conflict Test Results** list on the [Job Details Report](#).



AdminStudio®

[Back](#) | [Jobs List](#) | [Machine List](#) | [Summary of Access97](#) | [Help](#)

File Conflicts

Monday, August 29, 2005

Job Name:

Access97

Computer Name:

SCH01HOWEPXP

File Path	Conflicting Product	Publisher	Test Status	Resolution
C:\WINDOWS\system32\DRY5SRVR.HLP	RoboHelp Enterprise	eHelp Corporation	✖	ACE07
C:\WINDOWS\system32\msexc35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjet35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjet35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjint35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjint35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjter35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msjter35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE07
C:\WINDOWS\system32\mspdx35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msrd2x35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msrd2x35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msrepl35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msrtext35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\msxbse35.dll	RoboHelp Enterprise	eHelp Corporation	✖	ACE23
C:\WINDOWS\system32\ODBCINST.CNT	RoboHelp Enterprise	eHelp Corporation	✖	ACE07

1

2

For each test-ready package job, the following information is included:

Table 20-20 • Machine Result Summary Report Data

Item	Description
Job Name	Name identifying a test-ready package that was installed on this machine. The Job Identifier is entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard. By default, the Job Identifier is the package name. Click on this link to view the Job Details Report .
Computer Name	Unique identifier of an individual computer in the network.
File Path	Path and file name of installation file or component that caused the conflict.
Conflicting Product	Products that ACE errors were detected against.
Publisher	Manufacturer of conflicting product.
Test Status	Icon indicating whether the test-ready package successfully installed on the machine: Passed (✔) or Failed (✖),

Table 20-20 • Machine Result Summary Report Data (cont.)

Item	Description
Resolution	Links to the ACE conflicts topics, which describe the ACE conflicts and also explain how to resolve them.

Information on detected ACE conflicts is presented in the following topics:

- [Predeployment File Conflict Check: ACE07](#)
- [Predeployment File Conflict Check: ACE 23](#)
- [Predeployment Registry Check: ACE 10](#)
- [Predeployment Registry Check: ACE 24](#)
- [Resolving File Conflict Issues](#)

Predeployment File Conflict Check: ACE07

[ACE07](#) checks for the existence of the same file in components with different ComponentIds, and reports four types of errors, depending on the various combinations of source and target files and whether or not they originated in a merge module.

In Predeployment Test, component names are used to uniquely identify components across products, while in ConflictSolver, ACE07 uses the Component ID.

Predeployment File Conflict Check: ACE 23

[ACE23](#) identifies file duplication between source and target packages. ACE23 checks to see if files with the same name and destination directory have the same size, version, and language when comparing a source package against a target package. If a file with the same name and destination directory is found in both the source and target packages, but the file has a different size, version, or language, ACE23 fails.

Predeployment Registry Check: ACE 10

[ACE10](#) checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.

In Predeployment Test, component names are used to uniquely identify components across products, while in ConflictSolver, ACE10 uses the Component ID.

Predeployment Registry Check: ACE 24

[ACE24](#) checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value, ACE24 fails.

Resolving File Conflict Issues

In the File Conflict Results Page, the **Resolution** column lists the name of the ConflictSolver ACE for the detected failure, and a link to view ACE resolution information.

Predeployment Test Results By Machine Report

The Predeployment Test Results by Machine report lists all of the machines in your organization where test-ready packages have been installed. One way to access this report, is to by clicking the **Click here to see results by Machine** link on the [Predeployment Test Results By Job Report](#).

AdminStudio

[Back](#) | [Jobs List](#) | [Help](#)

PreDeployment Test Results by Machine

Monday, August 29, 2005

Results by Machine

Machine	IP Address	Operating System	Language	Total Jobs Run	Jobs Passed	Jobs Failed	Last Run
SCH01HOWEPXP	172.17.108.95	Microsoft Windows XP version 5.1 Service Pack 2 (Build 2600)	English	4	1	3	8/29/2005 1:33:00 PM
1							

[Click here](#) to see results by Job.


For each machine, the following information is included:

Table 20-21 • Predeployment Test Results by Machine Report Data

Option	Description
Machine Name	Name that identifies the machine on the network. Click this link to view the Machine Result Summary Report for this machine.
IP Address	Number that identifies the machine on the network.
Operating System	Operating system and version installed on the machine.
Language	Default language of operating system.
Total Jobs Run	Total number of test-ready packages that have been installed on the machine.
Jobs Passed and Jobs Failed	Number of test-ready packages that installed successfully on this machine and the number of test-ready packages that failed to install on this machine.
Last Run	Date and time that the last test-ready application was installed on the machine.
Click here to see results by Job.	Click to view the Predeployment Test Results By Job Report .
Jobs List	Click to view the Predeployment Test Results By Job Report .

Machine Result Summary Report




The Machine Result Summary report lists all of the test-ready package jobs run on a specific machine. To access this report, open the [Predeployment Test Results By Machine Report](#), and click on a Machine Name link.


AdminStudio®

[Back](#) | [Jobs List](#) | [Machine List](#) | [Help](#)

Machine Result Summary: [SCH01HOWEPXP]
Monday, August 29, 2005



Results by Machine

Job Name	File Name	User Name	Job Started	Job Finished	Result	Deployment Tests Failed	Conflict Tests Failed
Access97	Access97.msi	PatriciaHowe	8/29/2005 1:31:00 PM	8/29/2005 1:33:00 PM		1	2
AspectArchitect	AspectArchitect.msi	PatriciaHowe	8/29/2005 1:10:00 PM	8/29/2005 1:12:00 PM		0	2
Brio Navigator	Brio Navigator.msi	PatriciaHowe	8/29/2005 1:29:00 PM	8/29/2005 1:30:00 PM		0	1
DemoPlayer	DemoPlayer.msi	PatriciaHowe	8/29/2005 12:59:00 PM	8/29/2005 1:00:00 PM		0	0

1

For each test-ready package job, the following information is included:

Table 20-22 • Machine Result Summary Report Data

Item	Description
Job Name	Name identifying a test-ready package that was installed on this machine. The Job Identifier is entered on the Web Service URL Panel of the Predeployment Test Preparation Wizard. By default, the Job Identifier is the package name. Click on this link to view the Job Details Report .
File Name	Name of the original source .msi package.
User Name	Name of the user who installed the test-ready package.
Job Started and Job Finished	Date and time when the installation of the test-ready package began and ended.
Result	Icon indicating whether the test-ready package successfully installed on the machine: Passed () or Failed ()
Deployment Tests Failed	Total number of deployment tests that were included in the test-ready package that failed during installation. The tests that are included in a test-ready package are selected on the Deployment & Test Selection Panel of the Predeployment Test Preparation Wizard.
Conflict Tests Failed	Total number of conflict tests that were included in the test-ready package that failed during installation. The tests that are included in a test-ready package are selected on the Deployment & Test Selection Panel of the Predeployment Test Preparation Wizard.
Jobs List	Click to access the Predeployment Test Results By Job Report .
Machine List	Click to access the Predeployment Test Results By Machine Report .

Connection to New Catalog Page

You can choose to connect to a standalone Application Catalog database or the AdminStudio Enterprise Server Application Catalog database.

- [Connecting to a Standalone Application Catalog](#)
- [Connecting to the AdminStudio Enterprise Server Application Catalog](#)

Connecting to a Standalone Application Catalog



Task: *To connect to a standalone Application Catalog:*

1. Open the Predeployment Test Reports Web site.
2. On the **Predeployment Test Results by Job** page, click **Change Application Catalog**. The **Connection to New Catalog** page opens.
3. Open the **Standalone** tab.
4. Enter the name of the **Server** to which you want to connect.
5. For **How should SQL Server verify the authenticity of login ID?**, select one of the following options:
 - **With Windows NT authentication using the Network Login ID**—Use Windows network authentication when connecting to this database.
 - **With Database Server authentication using Login ID and password entered by user**—Use database server login identification for authentication when connecting to this database.
6. If you choose **Database Server authentication**, enter the **Login ID** and **Password**.
7. In the **Catalog Name** field, enter the name of an Application Catalog database available on the server.
8. Click **Save**.

Connecting to the AdminStudio Enterprise Server Application Catalog



Task: *To connect to a standalone Application Catalog:*

1. Open the Predeployment Test Reports Web site.
2. On the **Predeployment Test Results by Job** page, click **Change Application Catalog**. The **Connection to New Catalog** page opens.
3. Open the **Enterprise Server** tab. The address of the AdminStudio Enterprise Server is listed.
 - **If the AdminStudio Enterprise Server has been configured**, this address will be a URL to a running instance of an AdminStudio Enterprise Server.
 - **If the AdminStudio Enterprise Server has not yet been configured**, this address will be the default value of `http://localhost`.

4. If the **AdminStudio Enterprise Server site has not yet been configured**, or if the URL is incorrect, click the address link to open the **Select AdminStudio Enterprise Server URL** page, enter the correct URL, and click **Save**.
5. On the **Enterprise Server** tab, enter a valid **User Name** and **Password** for the specified AdminStudio Enterprise Server.
6. Click **Login**.

Manage Job Page

On the **Manage Job** page, you can change the **Job Status** of a job from **Failed** to **Passed** or delete the job from the Application Catalog.

If you click the **Manage Job** link on a **Job Summary Report** or **Job Details Report**, the **Manage Job** page opens. The following options are included:

Table 20-23 • Manage Job Page Options

Option	Description
Job Name	Name of the job that you are viewing.
MSI Name	Name and directory path of the test-ready package that was used to run this job.
Product Code	Product code of the test-ready package.
Job Status	<p>Status of this job:</p> <ul style="list-style-type: none"> • Failed • Passed <p>Marking a Failed job as being Passed does not modify any of the underlying test detail failure information that would normally be included in the Job Details Report or the Test Details Report. The original test results are always preserved for future reference, even though the predeployment test summary can be modified to appear as if the deployment tests succeeded.</p>
Override failed test	<p>Select this check box to change the status of a Failed job to Failed (Override to Passed). The job will then be listed in all other reports as Passed.</p> <p>If the Manage Job page was launched from a machine-specific report page, the name of this hyperlink is Mark test as Passed for this machine.</p>
Reason	Enter comments to document the reason the job's status was changed.
Save Changes	Click to save your selections and return to the report.

Table 20-23 • Manage Job Page Options (cont.)

Option	Description
Delete this job from the Application Catalog	Click to delete this job from the Application Catalog. If the Manage Job page was launched from a machine-specific report page, the name of this hyperlink is Delete the job data for this machine from the Application Catalog .

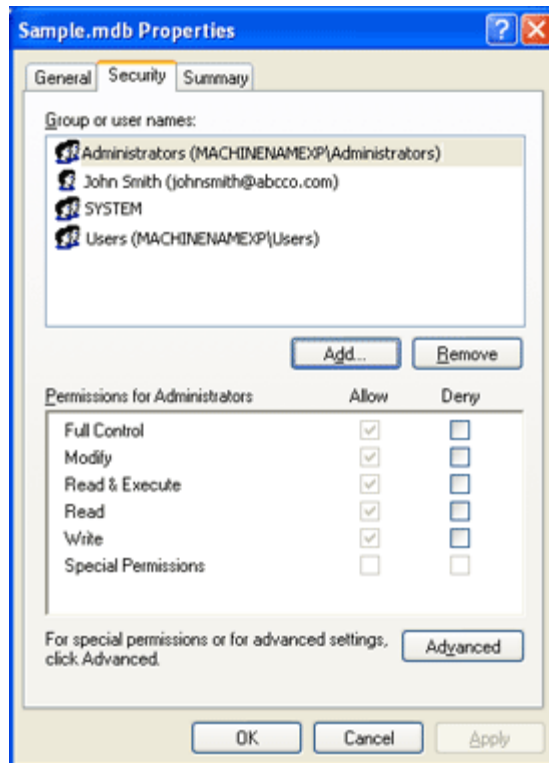
Assigning ASP.NET Permissions to Files or Folders

To use a Microsoft Access Application Catalog database on the Predeployment Test Results Web site, the database file must have ASP.NET permissions assigned to it. Files in the AdminStudio Shared directory inherit ASP.NET permissions from that directory. If you want to open a Microsoft Access database that is in another directory, you must assign ASP.NET permissions to that database file or the directory that it is in.

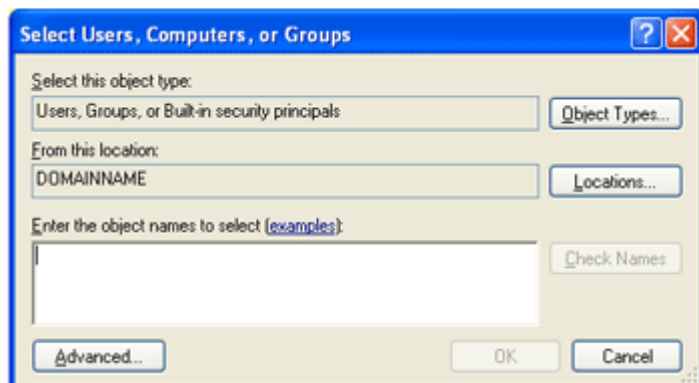


Task: *To assign ASP.NET permissions to a file or folder:*

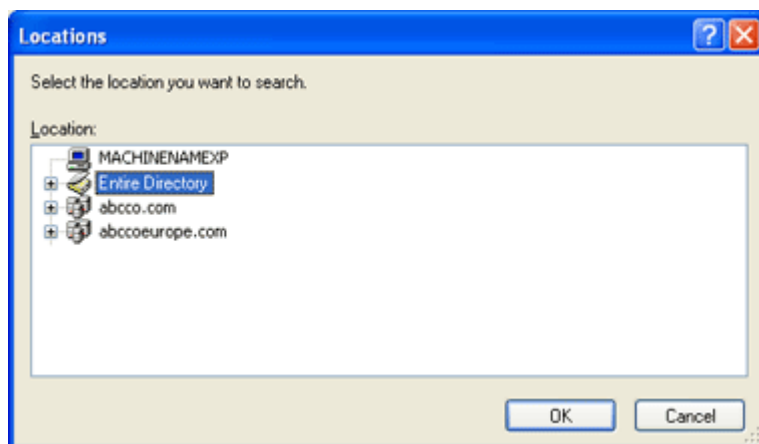
1. In Windows Explorer, select the file or folder that you want to assign ASP.NET permissions to select **Properties** from the context menu. The Properties dialog opens.
2. Click **Security** to open the **Security** tab.



3. Click **Add**. The **Select Users, Computers, or Groups** dialog box opens.



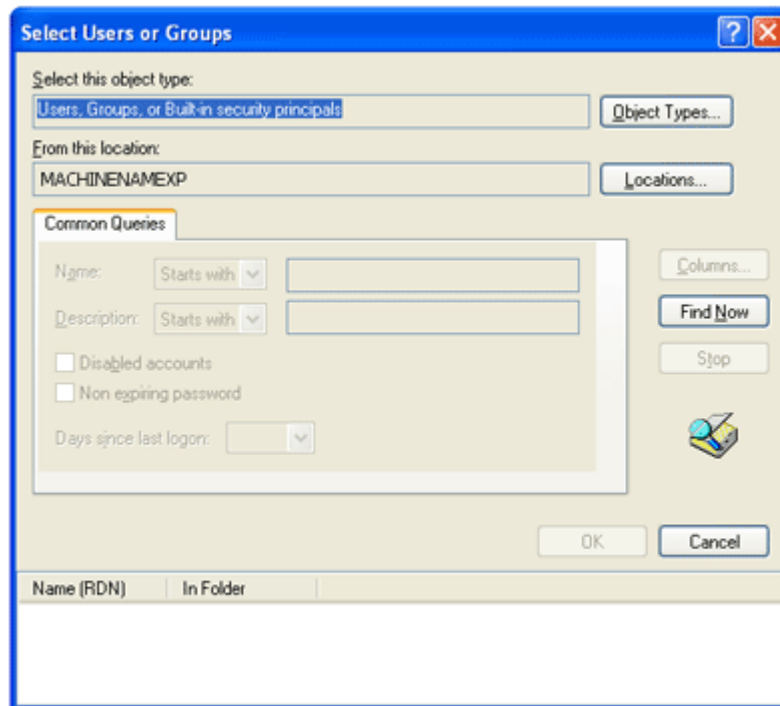
4. Click **Locations**. The **Locations** dialog box opens.



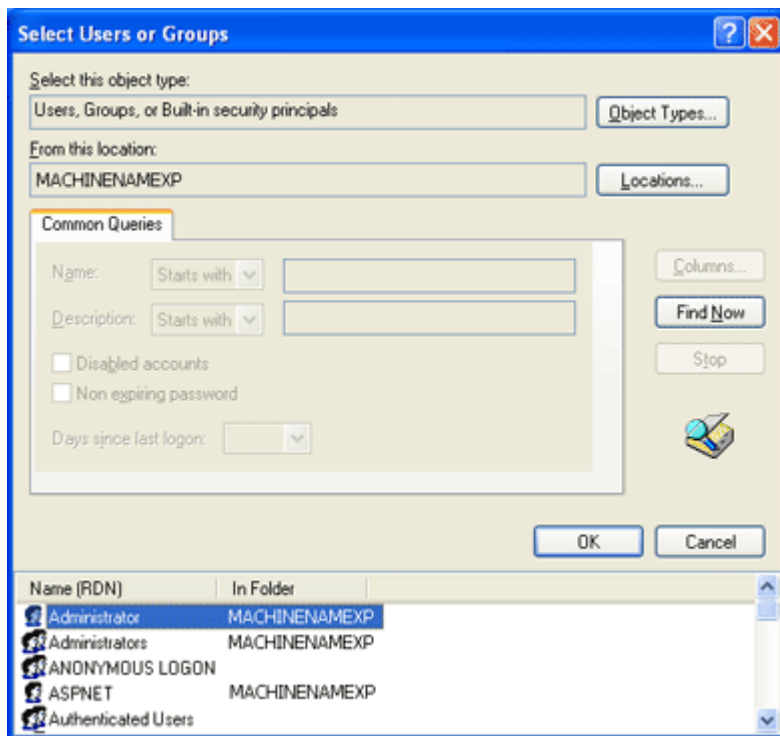
5. In the **Locations** list, select the name of your computer and click **OK** to return to the **Select Users or Groups** dialog box. The name of your computer is now listed in the **From this location** field.
6. In the **Enter the object names to select** list, enter **ASPNET** and click **Check Names**. The following is now listed in the **Enter the object names to select** list:

<YourComputerName>\ASPNET
7. If <YourComputerName>\ASPNET was found, continue to Step 7. If it was not found, perform the following additional steps:

- a. On the **Select Users or Groups** dialog box, click **Advanced**. The **Advanced** View of the **Select Users or Groups** dialog box opens.



- b. Click **Find Now**. All of the users and groups on your machine are listed.

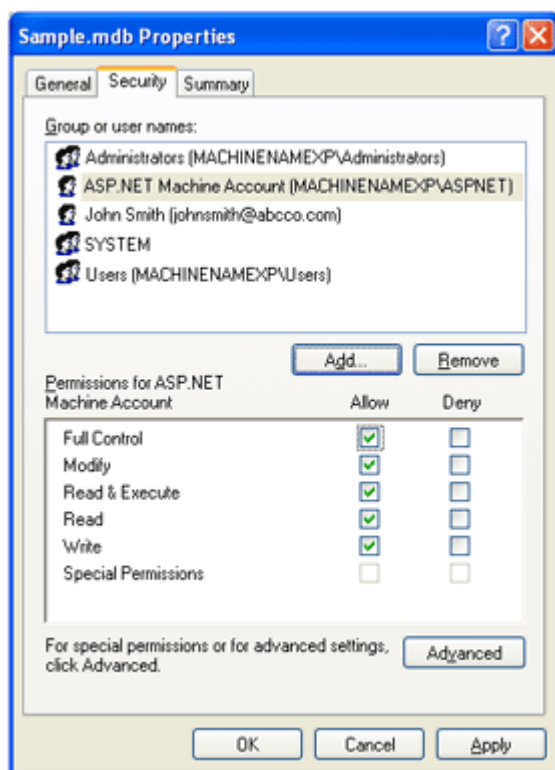


- c. In the **Name (RDN)** list, select **ASP.NET** and click **OK** to return to the **Select Users or Groups** dialog box.



Caution • If ASP.NET is still not listed, then ASP.NET is not installed on your machine. ASP.NET is automatically installed when you install AdminStudio. Re-install AdminStudio using the original installation CD.

8. Click **OK** to return to the **Security** tab of the **Properties** dialog box. ASP.NET Machine account is now listed in the **Group or user names** list.
9. In the **Group or user names** list, select **ASP.NET Machine account**, and then select **Full Control** in the **Allow** column in the **Permissions for ASP.NET Machine Account** list:



10. Click **OK** to exit the **Properties** dialog box.

ConflictSolver ACE Reference

The following ACEs are used by Predeployment Test to identify potential conflicts between installation packages.

Table 20-24 • ACEs Used by Predeployment Test

ACE Rule	Conflict Type	Brief Description of ACE
ACE07	Files	Checks for the existence of the same file in components with different ComponentIds.
ACE10	Registry	Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.
ACE23	Files	Identifies file duplication between source and target packages. ACE23 checks to see if files with the same name and destination directory have the same size, version, and language when comparing a source package against a target package. If a file with the same name and destination directory is found in both the source and target packages, but the file has a different size, version, or language, ACE23 fails.
ACE24	Registry	Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value, ACE24 fails.

ACE07

ACE07 checks for the existence of the same file in components with different ComponentIds. ACE07 reports four types of errors, depending on the various combinations of source and target files and whether or not they originated in a merge module:

Table 20-25 • Four Types of ACE07 Errors

Source File	Target File	Severity	CARD-Enabled	Conflict View Message	Output Message
Not Merge Module	Merge Module	Warning	No	The file [FILENAME] in the component [COMPONENTNAME] is identical to the merge module installed file in the component [COMPONENTNAME]. Confirm this error by running this package against ACE12.	The file [FILENAME] in the component [COMPONENTNAME] in the package [PACKAGENAME] is identical to the merge module installed file in the component [COMPONENTNAME] in the package [PACKAGENAME]. Confirm this error by running this package against ACE12.
Not Merge Module	Not Merge Module	Error	Yes	The file [FILENAME] in the component [COMPONENTNAME] is identical to a file in the component [COMPONENTNAME], but the components have different GUIDs.	The file [FILENAME] is identical in both the component [COMPONENTNAME] in the package [PACKAGENAME] and the component [COMPONENTNAME] in the package [PACKAGENAME], but the components have different GUIDs.
Merge Module	Merge Module	Error	No	The file [FILENAME] in the merge module installed component [COMPONENTNAME] is identical to a file in another merge module installed component [COMPONENTNAME], but the components have different GUIDs. Run ACE12 to determine which merge module is most appropriate to use.	The file [FILENAME] is identical and installed by merge modules in both the component [COMPONENTNAME] in the package [PACKAGENAME] and the component [COMPONENTNAME] in the package [PACKAGENAME], but the components have different GUIDs. Run ACE12 to determine which merge module is most appropriate to use.

Table 20-25 • Four Types of ACE07 Errors

Source File	Target File	Severity	CARD-Enabled	Conflict View Message	Output Message
Merge Module	Not Merge Module	Error	No	The file [FILENAME] in the merge module installed component [COMPONENTNAME] is identical to the file in the component [COMPONENTNAME]. Confirm this error by running the [PACKAGENAME] package against ACE12.	The file [FILENAME] in the merge module installed component [COMPONENTNAME] in the package %s is identical to the file in the component [COMPONENTNAME] in the package [PACKAGENAME]. Confirm this error by running the [PACKAGENAME] package against the ACE12.

Summary

Table 20-26 • ACE07 Summary

Conflict Type:	File
Description:	Checks for the existence of the same file in components with different ComponentIds.
Result:	ACE07 reports four types of errors, depending on the various combinations of source and target files and whether or not they originated in a merge module. (See table above.)
Resolution Type:	Automatic (CARD07) only if both the source and target files originated in a Merge Module. For all other errors, confirm the error by running the package against ACE12, and then, based upon the results, decide how to proceed.
Resolution:	The Source package ComponentId is changed to match the Target package Component Id.

ACE10

ACE10 checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds. The ACE10 Error String is displayed in the following format:

The registry entry [REGISTRY_ENTRY] [REGISTRY_KEY] in the component [COMPONENT1] in the package [PACKAGE1] conflicts with the same registry entry in the component [COMPONENT1] in the package [PACKAGE2].



Note • ACE10 is a warning, not an error.

Summary

Table 20-27 • ACE10 Summary

Conflict Type:	Registry
Description:	Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.
Result:	If the same Root/Key/Name registry combination is in more than one component, a warning is generated.
Resolution Type:	Manual
Resolution:	Change the ComponentId of the Source component to match that of the Target component.

Cause

The same Root/Key/Name registry combination is found in more than one component.

Conditions When an ACE10 Error Can Be Ignored

ACE10 uses data from the MSI Registry table to check for identical registry Root/Key/Name combinations in different components. However, there may be situations in which ACE10 will report an error unnecessarily. MSI supports a grammar for the Registry table in which if the Value field is preceded or terminated by the sequence tilde '[~]', then the registry value will be appended or prepended, respectively, to the existing registry Value. This sort of operation may be perfectly acceptable if the applications in question are modifying a common registry key in a manner consistent with its purpose.

You will need to decide individually if an ACE10 error is valid, but please consider checking the Registry Value field, as its contents may prove useful in helping you decide.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open the MSI package in InstallShield Editor.
2. Once the project is open, navigate to the Components View and find the component that needs to be modified by referring to the component name displayed in the error message.
3. In the Component's Property grid, change the value in the Component Code field to match the Component Code (ComponentId) of the component in the other project.

If it is unclear what the value should be changed to, follow these steps:

- a. Open up the package that will not be modified in Direct Edit mode by selecting Direct Editor under Additional Tools. A list of the package tables is displayed.
 - b. Select the Component table and search for the Component Name displayed in the conflict.
 - c. Once the entry is found, take note of the value displayed in the ComponentId column. This is the Component Code that should be used for the component in the other package.
4. Select Save As from the File menu and save the changes as a Windows Installer Transform (.mst) file.
5. Open ConflictSolver and reimport this package with its newly created transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE10 again.

ACE23

ACE23 identifies file duplication between source and target packages. ACE23 checks to see if files with the same name and destination directory have the same size, version, and language when comparing a source package against a target package. If a file with the same name and destination directory is found in both the source and target packages, but the file has a different size, version, or language, ACE23 fails.

The ACE23 Error String is displayed in the following format:

The file [EXECUTABLE_FILENAME, SIZE, VERSION, LANGUAGE] installed to location [PATHNAME] by component [COMPONENT1] in Package [PACKAGE1] conflicts with the same file in component [COMPONENT2] in Package [PACKAGE2].



Note • ACE23 can be used in conflict analysis when MSI Package, OS Snapshot File, or Marimba NCP File data types are selected as either source or target packages. However, if you specify only Marimba NCP or OS Snapshot files as your source packages, ACE23 and ACE24 will be the only ACEs that you can select on the **Conflict Options Panel** of the Conflict Wizard. All other ACEs will be unavailable.



Edition • Import support for Marimba Native Channel Packager (.ncp) files is included with the AdminStudio Enterprise Edition.

Summary

Table 20-28 • ACE23 Summary

Conflict Type:	Files
Description:	Checks to see if files with the same name and destination directory have the same size, version, and language.
Result:	If a file with the same name and destination directory is found in both the source and target package (and, in the case of an MSI package comparison, the packages have different ComponentId values), but the file has a different size, version, or language, ACE23 fails. If the Source and Target packages are MSIs and have the same Component Id value, no error is reported.
Resolution Type:	Manual
Resolution:	Investigate and decide which file has precedence. If the source file has precedence over the target file, remove the target file.

Cause

A file with the same name and destination directory is found in both the source and target package, but the file has a different size, version, or language.

Resolution

This conflict requires a manual resolution. Use ONE of the following solutions:



Task: *Replace the file in the source package with the same file found in the target package*

1. Retrieve a copy of the file from the operating system in which the target package was taken.
2. Open a transform file or MSI package in InstallShield Editor.
3. Once the project is open, navigate to the Files and Folders view, and find the file causing the conflict. Replace that file with the one retrieved from the operating system.
4. Select Save As from the File menu and save the changes as a Windows Installer Transform (.mst) file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE23 again.



Task: *Change the destination of the component that installs the file in the MSI package*

1. Open a transform file or MSI package in InstallShield Editor.
2. Once the project is open, navigate to the Components view, and select the component that contains the file causing the conflict.

In order to quickly find the component name, go to the File Table via the Direct Editor view and search for the file name. Then, check the component column for the component name.

3. Select the Destination field of the component from the Property grid and change it to a new destination.



Note • *Changing the destination may cause the application to break. Before changing the destination, verify that the application will still work using the new destination.*

4. Select Save As from the File menu and save the changes as a Windows Installer Transform (.mst) file.
5. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE23 again.

ACE24

ACE 24 checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value, ACE24 fails.

The ACE24 Error String is displayed in the following format:

The registry entry [REGISTRY_ENTRY] in [PACKAGE1] conflicts with the same registry entry in the [COMPONENT] in [PACKAGE2].



Note • *ACE24 can be used in conflict analysis when MSI Package, OS Snapshot File, or Marimba NCP File data types are selected as either source or target packages. However, if you specify only Marimba NCP Files as your source packages, ACE23 and ACE24 will be the only ACEs that you can select on the **Conflict Options Panel** of the Conflict Wizard. All other ACEs will be unavailable.*



Edition • *Import support for Marimba Native Channel Packager (.ncp) files is included with the AdminStudio Enterprise Edition.*

Summary

Table 20-29 • ACE24 Summary

Conflict Type:	Registry
Description:	Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value.
Result:	If a registry entry with the same registry hive, key, and value name in a package is found in both the source and target packages, but the registry entry has a different data type or value, ACE24 fails.
Resolution Type:	Manual
Resolution:	Investigate and decide which registry entry has precedence. If the source registry entry has precedence over the target registry entry, remove the target registry entry.

Cause

A registry entry with the same registry hive, key, and value name in a package is found in both the source and target package, but the registry entry has a different data type or value.

Resolution

This conflict requires a manual resolution. Follow these steps:



Task: *To resolve this conflict:*

1. Open a transform file or MSI package in InstallShield Editor.
2. Once the project is open, navigate to the Registry view, and find the registry value causing the conflict. Replace that value with the same registry value and data type found in the target package.
3. Select Save As from the File menu and save the changes as a Windows Installer Transform (.mst) file.
4. Open ConflictSolver and reimport this package with its transform file into your Application Catalog, and then use the Conflict Wizard to check it against ACE24 again.

License-Enabling Packages Using FLEXwrap

AdminStudio comes with two tools you can use to FLEX-enable packages:

- **FLEXwrap Assistant**—Allows you to place Windows Installer packages (.msi) that do not have embedded license management under the control of the FLEXnet software licensing technology for easy control and administration. See [Enabling License Tracking of Windows Installer Packages](#)
- **FLEXwrap Server Configuration Tool**—Allows you to monitor the usage of FLEX-enabled applications and manage FLEXwrap license files on the server. See [Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server](#).

You can also use the FLEXwrap Server Configuration Tool to FLEX-enable individual existing executable files (.exe). See [Using the FLEXwrap Server Configuration Tool to FLEX-Enable Executable \(.exe\) Files](#)

FLEXwrap user documentation is presented in the following sections:

Table 21-1 • FLEXwrap User Documentation

Section	Description
Enabling License Tracking of Windows Installer Packages	<p>Provides an overview of FLEXnet Software Licensing technology and how to use the FLEXwrap Assistant to inject licensing capabilities into existing Windows Installer packages. The following topics are included:</p> <ul style="list-style-type: none">• About FLEXnet Licensing Technology and FLEXwrap• Using the FLEXwrap Assistant• FLEXwrap Assistant Reference

Table 21-1 • FLEXwrap User Documentation (cont.)

Section	Description
Managing FLEXwrap License Files	<p>Explains how to manage FLEXwrap license files using the FLEXwrap Server Configuration Tool. The following topics are included:</p> <ul style="list-style-type: none">• Overview of the FLEXwrap License Server• Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server• Using the FLEXwrap Server Configuration Tool to FLEX-Enable Executable (.exe) Files• FLEXwrap Server Configuration Tool Reference

Enabling License Tracking of Windows Installer Packages

This section provides an overview of FLEXnet Software Licensing technology and how to use the FLEXwrap Assistant to inject licensing capabilities into existing Windows Installer packages. The information is presented in the following sections:

Table 21-2 • FLEXwrap Assistant User Documentation

Section	Description
About FLEXnet Licensing Technology and FLEXwrap	Describes the FLEXnet Software Licensing technology and provides an overview of the two components of FLEXwrap: the FLEXwrap Assistant, and the FLEXwrap Server Configuration Tool.
Using the FLEXwrap Assistant	Provides complete instructions on how to FLEX-enable Windows Installer packages using the FLEXwrap Assistant.
FLEXwrap Assistant Reference	Contains a description of each Page and Dialog Box in the FLEXwrap Assistant.

About FLEXnet Licensing Technology and FLEXwrap

The FLEXnet software licensing management platform is the de facto industry standard for the electronic licensing of software. The software operates on over forty operating systems.

Network license administrators can use FLEXwrap to inject FLEXnet software licensing technology into existing Windows Installer (.msi) packages so that they can track usage of the executables in those packages.

Once you have distributed a FLEX-enabled package to the users in your enterprise, you can manage the licensing of that application using [FLEXnet Manager](#).

Benefits of FLEXnet Software Licensing Technology

By using FLEXnet software licensing technology, you can significantly reduce the cost of your software investment by being able to control and monitor the usage of the software licenses on your network.

- **Track usage**—You can track concurrent usage of applications within the enterprise. You can track a list of specific users within the enterprise, and can trace users to a specific individual and machine. You can then use [FLEXnet Manager](#) to generate the following reports:
 - Concurrent usage
 - Installations
 - Report based on users and machine names
- **Enforce application license limits**—You can set the number of concurrent users permitted to run an application equal to the number of licenses that you have purchased, enabling you to comply with the terms and conditions of your software license agreements.
- **Enable overdraft licenses during peak usage times to determine accurate license requirements**—You can choose to allow users to run a software package even after its license limit has been reached, and then generate a report on actual usage. You could then purchase any additional licenses that are required, enabling you to only purchase the number of licenses that your organization actually needs.
- **Enable application usage during server down time**—You can choose to allow the wrapped application to run without a license when the FLEXwrap License Server is down or cannot be contacted, providing uninterrupted access to your wrapped applications.
- **Specify handling options for concurrent use**—Instead of requiring a license for each concurrent user (the default setting), you can choose to change this restriction by selecting a different grouping option to determine license requirements. You can specify that any of the following groupings would require only one license:
 - **Host**—All concurrent requests from the same computer
 - **User**—All concurrent requests from the same user
 - **User + Host**—All concurrent requests from the same user on the same computer
 - **Site**—All concurrent requests from the same site
- **Set license expiration date**—You can specify a date when all licenses for an application would expire.
- **Queuing behavior**—When the license limit is reached, you can specify whether the application would launch automatically when a license becomes available, ask the user if he wants to wait for an available license and display a list of current users, or inform the user that he should attempt to run the application at a later time.
- **Flexible instrumentation**—Packages can be FLEX-enabled either before or after repackaging or customization.

Types of Applications that Can Be FLEX-enabled

FLEXwrap can be used with:

- Internally developed applications
- Applications without built-in license enforcement
- Applications with a built-in, non-FLEXnet license manager



Note • FLEXwrap does not replace a built-in license manager. If you wrap an application that has a built-in license manager, the application must obtain a license from FLEXwrap in order to try to ask for a license from its built-in license manager.

However, if an application already contains a licensing mechanism, we do not recommend license-enabling it using FLEXwrap.

About FLEXwrap License Files

A FLEXwrap license file is a specialized license text file containing the applications under FLEXwrap management. This file also contains information about where the License Server runs, all the licensing settings, and the location of the log and options files. The license file (.lic) is read by the License Server and, when the **Allow application to run if the license server is unavailable** option is selected on the [Configure Connection Page](#), by the Wrapper.

Evaluation License File

When first using FLEXwrap Assistant, you may want to use a sample evaluation license file provided by AdminStudio.

To obtain an evaluation license file, click the **Request a FLEXwrap server license** link on the [Configure Connection Page](#) to open a [request form](#) on the AdminStudio Web site. Fill out the requested information and submit the form. You will then receive an evaluation license file named eval.lic via email.

When you receive this file, copy it to the following location:

AdminStudio Installation Directory\FLEXwrap\Server

Then when you choose this file on the [Configure Connection Page](#), the [FLEXwrap Server Name Dialog Box](#) opens and you are prompted to enter the machine name of the computer where the FLEXwrap server is installed.



Note • The Evaluation License File has a license limit count of **5**, which means that only five licenses are available for an application that is FLEX-enabled using this license.



Note • The evaluation license file (*eval.lic*) is for evaluation purposes only; you are not permitted to use it in a production environment. Also, the evaluation license file can only be used for a limited amount of time before expiring.

Production License Files

You purchase production license files from AdminStudio Support. To install a production license file, you first purchase license file credentials from AdminStudio Support, and then create the license file by entering the license file credentials on the **Certificate** tab of the FLEXwrap Server Configuration Tool. See [Installing Your Production FLEXwrap License File](#).

Selecting the Executable File to Wrap

On the **Select Files** page of the FLEXwrap Assistant, you are prompted to select a Windows Installer (.msi) package, and then to select the executable (.exe) files from that Windows Installer package that you want to license inject.

A Windows Installer package often has multiple executable files associated with it, so it may be unclear as to which of those files you should license inject. By default, the **Only show executable files with associated components with shortcuts option** is selected, which filters the list to show only those executables which can be launched separately using a Shortcut.

For example, if you were license-injecting a Windows Installer package that installs a suite of applications such as: wordprocessing.exe, spreadsheet.exe, presentation.exe

you would need to license-inject all three of those executable files in order to enforce license management.

But if only one of a Windows Installer package's executable files launches separately (and all the others are called by the main executable to run behind the scenes to perform specific functions), you would only have to license-inject the main, launchable executable file.

You also have the option of clearing the selection of any executable. For example, if an executable is a command line utility that you do not want to enforce licensing on, you can just clear the check box for that executable.

About Output Folders

The **Output folder** on the [Finalize Package Page](#) defaults to the folder specified for **Default Output Folder** on the [Options Dialog Box](#).

When the selected package is wrapped, FLEXwrap Assistant creates a subfolder in the default output folder and gives it the same name as the Windows Installer file. Then FLEXwrap Assistant copies the Windows Installer file and all of its associated files and subfolders to this new subfolder before starting the wrapping process.

For example, if you were license enabling a Windows Installer package named Graph2005.msi, and if the **Output folder** on the [Finalize Package Page](#) was set to:

C:\WrappedPackages

the selected Windows Installer file and all of its associated files and subfolders would be copied to:

C:\WrappedPackages\Graph2005






How FLEXwrap Assistant Works

The FLEXwrap Assistant, a tool in the AdminStudio Tools Gallery, is used to FLEX-enable Windows Installer packages. After a Windows Installer package is wrapped to include the licensing technology, the output is still a Windows Installer package which has full functionality and manageability, with all Windows Installer capabilities intact.

Actions You Take

When you use the FLEXwrap Assistant to inject licensing capabilities into a Windows Installer package, you perform the following steps:

Table 21-3 • Wrapping a Windows Installer Package

Icon	Step	Description
	Step 1: Select Files	Identify the executables (.exe files) in a Windows Installer package (.msi file) that you want to wrap, and specify whether to encrypt the original executables to prevent end users from bypassing licensing limits.
	Step 2: Set Limits	Specify the number of concurrent licenses available for this application, the number of overdraft licenses available for peak usage, and the action the application will take if no licenses are currently available.
	Step 3: Define Access	Defining which requests or group of requests require a separate license. You can also restrict application usage by specific computers.
	Step 4: Configure Connection	Configure a connection to the FLEXwrap License Server.
	Step 5: Finalize Package	Specify an output folder and initiate the creation a new FLEX-enabled Windows Installer package.

Actions Taken by the FLEXwrap Assistant

The following diagram details the main actions performed by the FLEXwrap Assistant when you wrap a Windows Installer package:

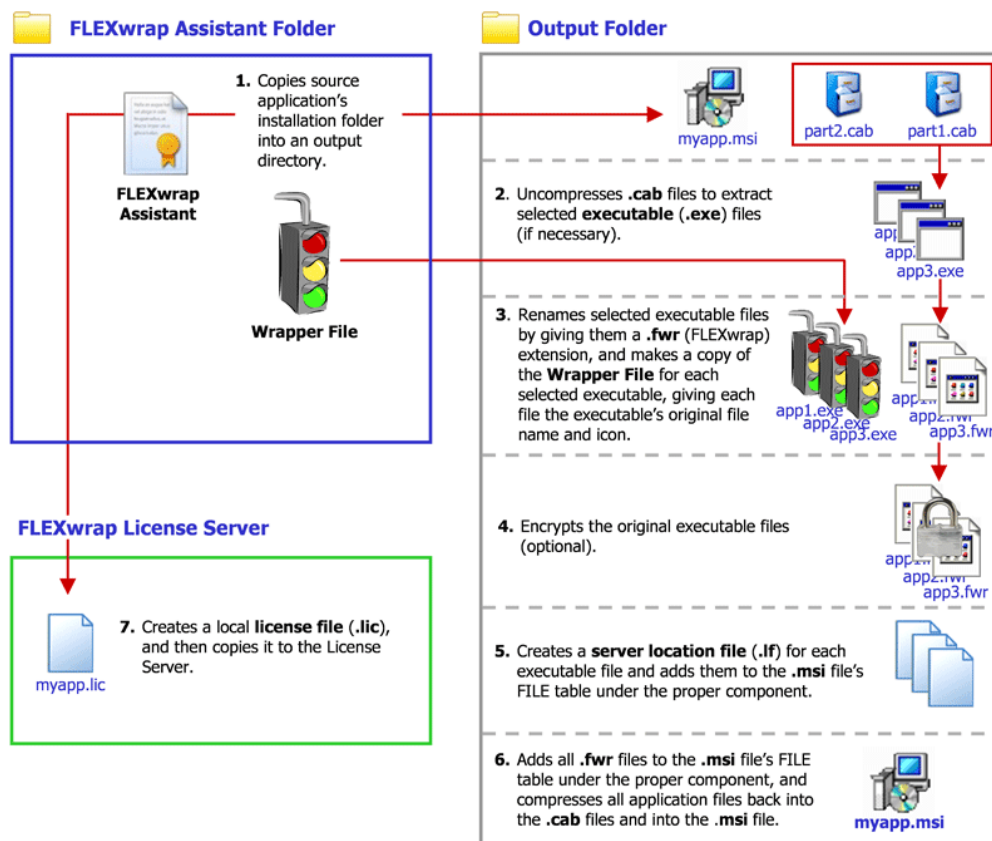


Figure 21-1: FLEXwrap Assistant Overview

When you click **Build Package** on the **Finalize Package Panel**, the FLEXwrap Assistant takes the following actions:

- 1. Copies package to output folder**—Copies the source Windows Installer folder to a subfolder of the output folder you specified.
- 2. Extracts executables**—Uncompresses this package's .cab files (if any) to extract the executables (.exe files) that you selected.
- 3. Renames executables and replaces them with Wrapper**—Renames selected .exe files to .fwr, and makes a copy of the copy of the Wrapper file for each selected .exe file, giving each file the executable's original file name and icon.
- 4. Encrypts executables**—Encrypts the original executables, if requested.
- 5. Specifies location of License Server**—For each executable, the FLEXwrap Assistant creates a location file (.lf) containing the location of the License Server, and adds the .lf files to the .msi File table under the proper component.

6. **Recompresses Windows Installer package**—Adds all .fwr files to the package's File table under the proper component, and compresses all application files back into the .cab files and into the .msi.
7. **Creates license file and copies it to the License Server**—Creates a local license file (myapp.lic) for this application, and then copies it to the License Server.

Launch Process for a Wrapped Application

When a user attempts to launch a wrapped application, the following occurs:

- **Wrapper launches**—When a user chooses to open an application, the Wrapper launches (which is transparent to the user).
- **Wrapper obtains location of License Server**—The Wrapper obtains the location of the License Server from the server location file (.lf).
- **Wrapper contacts License Server to obtain a license**—The Wrapper attempts to obtain a license for the wrapped application from the License Server.
- **Wrapper launches the wrapped application**—One of the following occurs:

If this is true ...	BUT this is also true,	then this happens
License is obtained		Wrapper runs the wrapped application's binary.
No license is obtained	Enable overdraft licenses option on the Set Limits Page is selected	Wrapper runs the wrapped application's binary.
License Server cannot be contacted	Allow application to run if the License Server is unavailable option on the Configure Connection Page page is selected	Wrapper runs the wrapped application's binary.

If this is true ...	BUT this is also true,	then this happens
No license is obtained	Neither of the above options was selected.	<p>Wrapper follows the option that was selected under Action to take when license limit is reached on the Set Limits Page page:</p> <ul style="list-style-type: none"> • Ask User—A list of current license users opens, and the user is prompted to select whether he wants to wait for a license to become available. • Always Queue—The user is added to a waiting list and then the application is automatically launched when a license becomes available. • Never Queue—Application is not launched and the user is instructed to attempt to launch the application later.

License checkouts are nearly instantaneous, so users should not notice any delay in starting wrapped applications.

- **Usage information written to report log**—If a license is obtained and if report logging is enabled, license usage information is written to the FLEXwrap report log. You could then use FLEXnet Manager reporting to generate usage reports.

The following diagram illustrates the activities that occur when a wrapped application is launched:

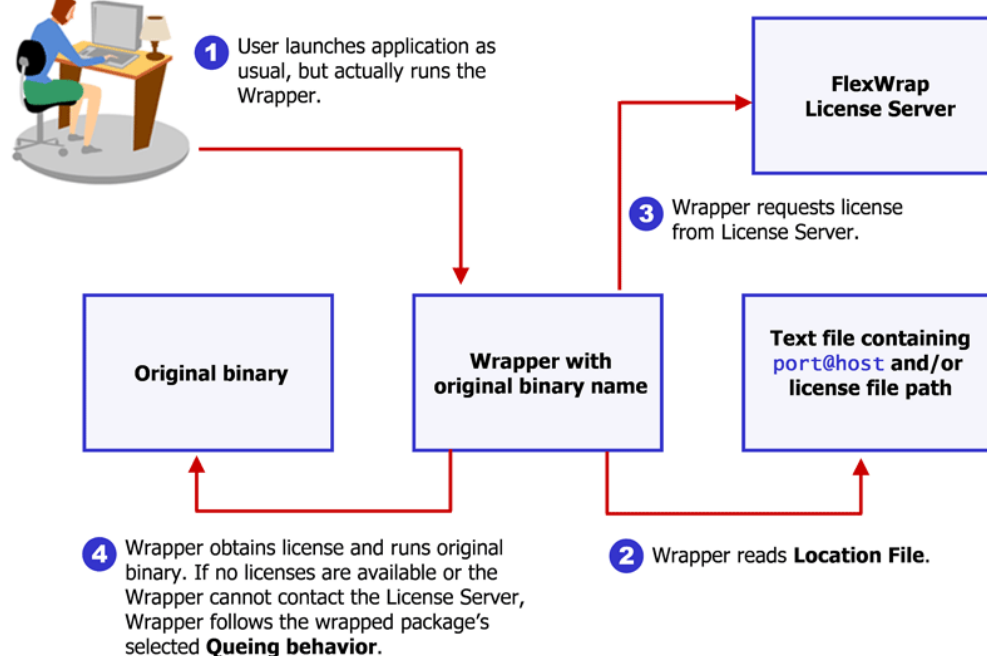


Figure 21-2: Diagram of activities that occur when a wrapped application is launched



Note • This diagram assumes that no overdraft licenses were provided and the application is not permitted to run if the License Server cannot be connected

Using the FLEXwrap Assistant

The procedures listed in this section explain how to use the FLEXwrap Assistant to license-enable Windows Installer packages, and describes the various licensing options.

- [Launching the FLEXwrap Assistant](#)
- [License-Enabling a Windows Installer Package](#)
- [Setting Licensing Options](#)

Launching the FLEXwrap Assistant

The FLEXwrap Assistant can be launched from AdminStudio from the following locations:

- Click the **FLEXwrap Assistant** icon in the Tools Gallery on the AdminStudio 9.5 Start Page.
- Click the **FLEXwrap Assistant** icon in the Tools Gallery on the AdminStudio Tools Tab.

- Click the **Prepare Application for Usage Management** link in the Application Tasks list on the AdminStudio Start Page.
- Click the **Prepare Application for Usage Management** link in the More Options list on the Distribute Page of the Process Assistant.
- On the Windows **Start Menu**, point to **All Programs, AdminStudio, AdminStudio 9.5 Tools, FLEXwrap**, and click **FLEXwrap Assistant**.

License-Enabling a Windows Installer Package

To license-enable a Windows Installer package, perform the following tasks:

- [Selecting a Windows Installer Package to License-Enable](#)
- [Setting License Limits and Queuing Behavior](#)
- [Defining License Requirements and Access Restrictions](#)
- [Configuring a Connection to the FLEXwrap License Server](#)
- [Specify the Output Folder and Build Package](#)

Selecting a Windows Installer Package to License-Enable

The first task is to select the Windows Installer package that you want to FLEX-enable, and specify whether to encrypt the original executables to prevent end users from bypassing licensing limits.



Task: *To license-enable a Windows Installer package:*

1. Launch the FLEXwrap Assistant. The **FLEXwrap Home Page** opens.
2. Click the **Select Files** icon in the navigation bar. The **Select Files Page** opens.
3. Click the **Browse** button and select the Windows Installer (.msi) package file that you want to license-enable.

The names of all of the executables contained the selected Windows Installer package are now listed in the **Select Executable File(s) to License-Inject** list, and by default all are selected.
4. Often a Windows Installer package contains multiple executable (.exe) files, but usually only a subset of these executables have a Windows shortcut associated with it. If the **Only show executable files with associated components with shortcuts** option is selected, only those executables that the Windows Installer package creates a shortcut for are listed. To view a list of all executables regardless of Shortcut status, clear this option.
5. Clear the selection of any executables that you do not wish to license-enable. You can use the **Select All** or **Clear All** buttons to quickly select or clear selections.
6. If more than one executable is selected, the **Treat selected executable files as a suite** option is enabled. If you want to license-enable these executables as a suite, meaning that only one license is required to launch multiple executables from this package, select this option and enter a **Suite name**.

For more information, see [Benefits of Using the Suite Option](#).

7. If you want to prevent users from running non-license-injected versions of the selected file(s) by renaming and using the original executable, select the **Encrypt original executable file(s)** option.

See [Benefits of Encrypting the Source Executables](#) for more information.

8. Click the **Set Limits** icon in the navigation bar. The **Set Limits Page** opens.
9. Proceed with [Setting License Limits and Queuing Behavior](#).

Setting License Limits and Queuing Behavior

In this task you specify the number of concurrent licenses available for this application, the number of overdraft licenses available for peak usage, and the action the application will take if no licenses are currently available.



Task: *To set license limits and overdraft behavior:*

1. After you have performed the steps described in [Selecting a Windows Installer Package to License-Enable](#) and the **Set Limits Page** is open, enter the number of concurrent licenses available for this application in the **License limit** box. For help in determining your license limit, see [Setting Your License Limit](#).

2. If you want to enable users to continue to use this application even when the license limit has been reached, select the **Enable overdraft licenses** option and enter a value in the **Overdraft limit** box.

See [Using Overdraft Support to Determine Application Usage](#) for a description of your overdraft support options.

3. If you want this application's licenses to expire on a specific date, select the **Set an expiration date for the licenses** option and select a date from the **Licenses expires on** list. If you do not want these licenses to expire, leave this option unselected.

See [Setting a License Expiration Date](#) for more information.

4. Under **Queueing behavior**, select one of the following options:

- **Ask user**—If all licenses are in use and another user attempts to launch the application, a list of the current license users opens, and the user is prompted to select whether he wants to wait for a license to become available (when one of the current license users closes the application). If he selects this option, the application will launch when a license becomes available.

By displaying a list of current license users, the user who is waiting has the option of contacting one of the license users to see if they will close the application to make a license available.

- **Always queue**—If all licenses are in use and another user attempts to launch the application, always add the user to a waiting list and then launch the application when a license becomes available.
- **Never queue**—If all licenses are in use and another user attempts to launch the application, inform user to attempt to launch the application later.

For more information, see [Choosing the Right Queuing Option for Your Application](#).

5. Click the **Define Access** icon in the navigation bar. The **Define Access Page** opens. Proceed with [Defining License Requirements and Access Restrictions](#).

Defining License Requirements and Access Restrictions

In this task you define which requests or group of requests require a separate license. You can also restrict application usage by specific hostids (computer names).



Task: *To define license requirements and set access restrictions:*

1. After you have performed the steps described in [Setting License Limits and Queuing Behavior](#) and the **Define Access Page** is open, select one of the following license requirement grouping options:
2. You can change this restriction by selecting one of the other four grouping options:
 - **Process**—Each concurrent instance of a license-enabled application requires a separate license. (Default)
 - **Host**—All concurrent requests from the same computer use only one license, regardless of whether multiple users are accessing the application from the same computer.
 - **User**—All concurrent requests from the same user use only one license, regardless of whether the same user is accessing the application from multiple computers.
 - **User + Host**—All concurrent requests from the same user on the same computer use only one license. This is the default selection if grouping concurrent license requests.
 - **Site**—All concurrent requests use the same license.

For more information, see [Grouping Options for Concurrent Use](#).

3. A **hostid** is the address of a computer in a TCP/IP network that uniquely identifies it. To restrict application usage by hostid, select the **Restrict access by hostids listed below** option, and perform the following steps to add hostids to the **Permitted hostids** list:
 - a. Click **Add**. The **Permitted Hostid** dialog box opens.
 - b. In the **Hostid Type** list, select the type of hostid that you want to restrict usage by: **Ethernet address**, **Internet IP address**, **Machine name**, **User name**, **Windows disk serial number**, or **Other**.
 - c. In the **Hostid Value** text box, enter a value for the selected **Hostid Type**.

To determine a machine's hostid, launch **FLEXwrap LMTTOOLS** on that machine, and open the **System Settings** tab:

Chapter 21: License-Enabling Packages Using FLEXwrap

Enabling License Tracking of Windows Installer Packages

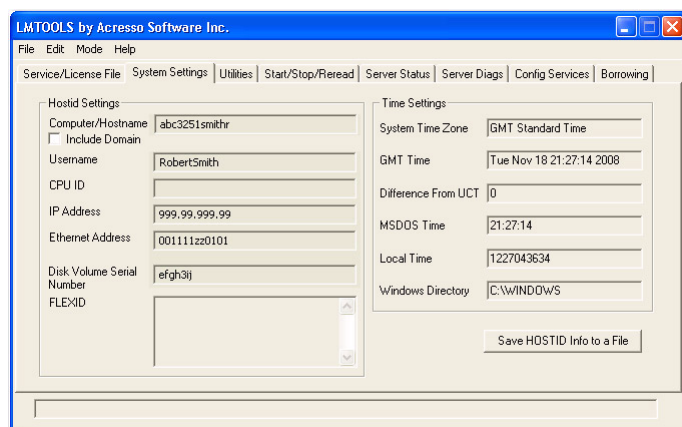


Figure 21-3: Hostid information on the Systems Settings tab of FLEXwrap LMTOOLS

- d. Click **OK**. The **Permitted Hostid** dialog box closes and the hostid that you just defined in now listed in the **Permitted hostids** list.
- e. If you want to edit an existing hostid, select an item in the **Permitted hostids** list and click **Edit**

For more information, see [About Hostids](#).

4. Click the Configure Connection icon in the navigation bar. The Configure Connection Page opens. Proceed with [Configuring a Connection to the FLEXwrap License Server](#).

Configuring a Connection to the FLEXwrap License Server

In this task you configure a connection to the FLEXwrap License Server.



Task:

To define license requirements and set access restrictions:

1. After you have performed the steps described in [Defining License Requirements and Access Restrictions](#) and the **Configure Connection Page** is open, do one of the following:

If you already have a license file, perform the following steps:

- a. Click **Browse** and select a **FLEXwrap server license (.lic) file**. This file is not modified and is only used to get your FLEXwrap License Server settings.
- b. After you select a license file, the **Host name** and **Port** fields are automatically populated with the information in the license file.

If you do not have a license file, perform the following steps to obtain and open an evaluation license file:

- a. Click the **Request a FLEXwrap server license** link to open a [request form](#) on the AdminStudio Web site. Fill out the requested information and submit the form. You will then receive an evaluation license file named `eval.lic` via email.
- b. Copy the `eval.lic` file to the following location:

AdminStudio Installation Directory\FLEXwrap\Server

- c. On the **Configure Connection Page**, click **Browse** and select the file that you just obtained: eval.lic. The FLEXwrap Server Name dialog box opens, prompting you to enter the machine name of the computer where the FLEXwrap server is installed.
- d. Enter a machine name and click OK.



Note • The Evaluation License File has a license limit count of **5**, which means that only five licenses are available for an application that is FLEX-enabled using this license.

2. If you want to be able to specify the server location when this application is installed, select the **Allow host name and port redefinition at install time** option.

If this option is selected, FLEXwrap injects a custom action into the .msi file that enables you to set the server name and port from the command line. For example:

```
msiexec /i "<path_to_msi_file>\wrapped.msi" SWRAP_LICENSE_FILE=9987@testserver
```

You would choose this option if you wanted the flexibility to reassign the FLEXwrap License Server host name/ port at install time for any reason, such as if you wanted to install the same wrapped application to multiple servers across different locations at the same organization.

3. To allow this wrapped application to be able to launch even when the FLEXwrap License Server is down or is having communication problems, select the **Allow application to run if the License Server is unavailable** option. This option stores the license file on the local machine. By default, this is unselected and is not recommended because it may enable users to run unlicensed versions of an application, circumventing the protection that FLEXwrap provides.
4. If you want to allow an application to run even when the License Server becomes unavailable, select the **Specify timeout value** option. Then, in the **Timeout value** box, enter the number of minutes that the connection has to be lost before the application is closed.
 - By default, this option is not selected and therefore there is no timeout period. The application will continue to run without a license and will continue to attempt to regain its license indefinitely.
 - If this option is selected and a **Timeout value** is entered, the application warns the user and then exits if the license cannot be regained within the timeout period.
5. Click the **Finalize Package** icon in the navigation bar. The **Finalize Package Page** opens. Proceed with [Specify the Output Folder and Build Package](#).

Specify the Output Folder and Build Package

In this task, you specify an output folder and initiate the creation a new FLEX-enabled Windows Installer package.



Task: *To specify the Output folder and enable licensing:*

1. After you have performed the steps described in [Configuring a Connection to the FLEXwrap License Server](#) and the **Finalize Package Page** is open, you can click **Browse** and select the **Output folder** where you want the wrapped application to be stored or you can just accept the default location.
2. If you want to automatically update the package code of this wrapped package to differentiate it from the original package, select the **Update package code** option. By default, this option is selected. See [When to Update the Package Code](#) for more information.
3. If you want to add a comment to this package to indicate that it is a wrapped package or to provide information on next steps, select the **Modify package comments** option and enter comments in the box. These comments will be visible from Windows Explorer if you right click on this .msi file and select **Properties** from the context menu. See [Purpose of Packaging Comments](#) for more information.
4. Click **Build Package** to create a new Windows Installer package with built-in licensing and store it in the directory that you specified.

Next Steps

After you have license-enabled a package, the next step is to restart your FLEXwrap License Server or issue a reread command. For instructions on restarting your License Server, see [Starting the License Server](#).

You also may want to use FLEXnet Manager and set up usage reporting so that you can track concurrent usage of this application within the enterprise. You can track a list of specific users within the enterprise, and can trace users to a specific individual and machine. For more information on FLEXnet Manager's reporting features, see [FLEXnet Manager](#). After you have FLEX-enabled a package, you cannot open up the wrapped package in the FLEXwrap Assistant to edit any of the settings. Instead, select the original Windows Installer file and follow the [License-Enabling a Windows Installer Package](#) procedure to create another wrapped version of the package.

Setting Licensing Options

When using the FLEXwrap Assistant to license-enable a Windows Installer package, you can use the following licensing options:

- [Benefits of Using the Suite Option](#)
- [Benefits of Encrypting the Source Executables](#)
- [Setting Your License Limit](#)
- [Using Overdraft Support to Determine Application Usage](#)
- [Setting a License Expiration Date](#)
- [Choosing the Right Queuing Option for Your Application](#)

- [Grouping Options for Concurrent Use](#)
- [Setting Licensing Behavior During Server Down Time](#)
- [Setting Timeout Periods](#)

Benefits of Using the Suite Option

If a Windows Installer package contains multiple executables, you can specify that you want all of the executables to belong to the same suite and therefore require only one license.

To do this, select multiple executable files on the [Select Files Page](#), select the **Treat selected executable files as a suite** option, and specify a **Suite name**.

For example, if a Windows Installer package named GraphicsUtilities.msi includes a drawing application (draw.exe), a screen capture application (capture.exe), and a page layout application (layout.exe), you can choose to treat these applications as a suite so that the user would only have to purchase one license to activate all applications in the suite.

Benefits of Encrypting the Source Executables

When the FLEXwrap Assistant wraps a Windows Installer package, it renames the selected .exe files to .fwr, and makes a copy of the Wrapper file for each selected .exe file, giving each file the executable's original file name and icon. All of these files are compressed back into the Windows Installer .msi file.

Because the original, unwrapped executables are installed along with the wrapped executables, it is possible for a user to rename the executable files with the .fwr extension back to having an .exe extension, and to launch the original, unwrapped executables and bypass the licensing limits you have set.

Therefore, if you want to prevent users from running non-license-injected versions of the wrapped executables, select the **Encrypt original executable file(s)** option on the [Select Files Page](#). This would make it impossible for the original executable files to run.



Caution • This encryption feature of FLEXwrap is not guaranteed because some applications can be configured to discourage any alteration, and encryption of the application binary may not be allowed.

Setting Your License Limit

On the [Set Limits Page](#), you should set the **License limit** (the number of concurrent users permitted to run an application) equal to the number of licenses that you have purchased, enabling you to comply with the terms and conditions of your software license agreements.

However, the FLEXwrap Assistant provides several other options you can use to manage access to an application while still complying with your license agreements:

- **Overdraft support**—By selecting the **Enable overdraft licenses** option on the [Set Limits Page](#), you can choose to allow a specified number of users to run a software package during peak usage times even after its license limit has been reached, and then generate a report on actual usage. You could then purchase any

additional licenses that are required, enabling you to only purchase the number of licenses that your organization actually needs.

- **Grouping options**—By default, each application instance requires its own license. You can change this restriction by selecting a different grouping option so that license requests are grouped by computer, by user, by user and computer, or by site. For more information, see [Grouping Options for Concurrent Use](#).

Using Overdraft Support to Determine Application Usage

By selecting the **Enable overdraft licenses** option on the [Set Limits Page](#), you can choose to allow a specified number of users to run the software package during peak usage times even after the application's license limit has been reached.

You can then use FLEXnet Manager to generate a report on actual usage, enabling you to accurately determine your organization's license requirements, and purchase any additional licenses that are required. Implementing Overdraft support gives you the information you need to purchase only the number of licenses that your organization actually needs.

Setting a License Expiration Date

On the [Set Limits Page](#) of the FLEXwrap Assistant, you can choose to enter an expiration date for this application's licenses. You would want to set an expiration date if you were aware that your license agreement for this application expires on a specific date, and you want to make sure that no one in your organization can launch the application after that date. This enables you to comply with the terms and conditions of your software license agreement.

If you do not have plans to let this application's license agreement expire, do not select this option.

Choosing the Right Queuing Option for Your Application

On the [Set Limits Page](#) under **Action to take when license limit is reached**, you can select one of the following options for the selected package:

- **Ask user**—If all licenses are in use and another user attempts to launch the application, a list of the current license users opens, and the user is prompted to select whether he wants to wait for a license to become available (when one of the current license users closes the application). If he selects this option, the application will launch when a license becomes available.

By displaying a list of current license users, the user who is waiting has the option of contacting one of the license users to see if they will close the application to make a license available. You might want to choose this option if most of the users who would be using this package know each other and contact each other on a frequent basis.

- **Always queue**—If all licenses are in use and another user attempts to launch the application, always add the user to a waiting list and then launch the application when a license becomes available. You might want to choose this option if most of the users who would be using this package do not know each other and, therefore, probably would not contact any of the current users on the list.

- **Never queue**—If all licenses are in use and another user attempts to launch the application, inform the user to attempt to launch the application later. You might want to use this option when most of the people using this application usually use it for long periods of time (and therefore would not be likely to close it any time soon).

Grouping Options for Concurrent Use

By default, license requests are grouped by **Process**: each concurrent request to launch the application uses one license. Instead of requiring a license for each concurrent user, you can choose to change this restriction by selecting a different grouping option to determine license requirements. You can specify that any of the following groupings would require only one license:

- **Host**—All concurrent requests from the same computer.
- **User**—All concurrent requests from the same user.
- **User & Host**—All concurrent requests from the same user on the same computer.
- **Site**—All concurrent requests from the same site.

You should select the grouping option that concurs with your licensing agreement for this software package.

Setting Licensing Behavior During Server Down Time

By selecting the **Allow application to run if the license server is unavailable** option on the [Configure Connection Page](#), you can choose to allow the wrapped application to run without a license when the FLEXwrap License Server is down or cannot be contacted, providing uninterrupted access to this application. When this option is chosen, the license file is stored on the local machine.

You would not want to select this option if you were concerned that having it selected might enable some users to use this application without a valid license.

Setting Timeout Periods

You would select the **Specify timeout value** option on the [Configure Connection Page](#) to specify the number of minutes that the connection to the License Server has to be lost before the application is closed.

By default, the **Specify timeout value** option is not selected and therefore there is no timeout period. The application will continue to run without a license and will continue to attempt to regain its license indefinitely.

If the **Specify timeout value** option is selected and a **Timeout value** is entered, the application warns the user and then exits if the connection to the License Server cannot be regained within the timeout period. You would want to specify a timeout value if you were concerned that if no timeout value was specified, some users would be able to use this application without a valid license and violate your licensing agreement.

Setting Configuration Options

The following topics describe FLEXwrap Assistant configuration and communication concepts:

- [About Hostids](#)
- [Changing the Host Name or Port at Install Time](#)
- [When to Update the Package Code](#)
- [FLEX-enabling a Signed Package](#)
- [Purpose of Packaging Comments](#)

About Hostids

You can choose to restrict application access by hostid so that only permitted hostids would be able to use the application. A **hostid** is the address of a computer in a TCP/IP network that uniquely identifies it. The hostid could be the Ethernet address, IP address, machine name, user name, Windows disk serial number, or any other unique identifier for that machine.

To lock a license file to a specific group of machines, choose the **Restrict access by hostids listed below** option on the [Define Access Page](#) of the FLEXwrap Assistant, and add **Permitted hostids** to the list.

Changing the Host Name or Port at Install Time

If you want to be able to specify the server location when this application is installed, select the **Allow host name and port redefinition at install time** option on the [Configure Connection Page](#).

If this option is selected, FLEXwrap injects a custom action into the .msi file that enables you to set the server name and port from the command line. For example:

```
msiexec /i "<path_to_msi_file>\wrapped.msi" SWRAP_LICENSE_FILE=9987@testserver
```

You would choose this option if you wanted the flexibility to reassign the License Server host name/port at install time for any reason, such as if you wanted to install the same wrapped application to multiple servers across different locations at the same organization.

When to Update the Package Code

If you want to automatically update the package code of this wrapped package to differentiate it from the original package, select the **Update package code** option. By default, this option is selected.

You would want to select this option perhaps if you wanted to import both the wrapped and the original package into the AdminStudio Application Catalog, which requires the each package to have a different package code. If you only wanted to import the wrapped package into the Application Catalog, you would not need to select this option.

If the wrapped package used the same package code as the original package, it would cause issues with patches and anything else that would treat both packages the same. If patches intended to update the original application's binary were applied to the wrapped package, the patch would fail. By creating a new package code, this problem is avoided.

FLEX-enabling a Signed Package

Packages can include a digital signature, and those packages are referred to as signed packages. A signed package is identical to an unsigned package except for the digital signature.

If a package has a valid digital signature, this ensures that the package has not been modified since the signature was applied to the package. Using signed packages provides a secure method of downloading or installing packages because the digital signature can be verified before the package is added to a system.

If you choose to FLEX-enable a signed package, the following warning message appears and the digital signature is invalidated:

Warning: The Windows Installer package you selected is digitally signed.

If you choose to continue and use the FLEXwrap Assistant to modify this package, you lose this package's digital signature and the application can no longer be verified before download or installation. However, all of the package's functionality remains intact and it installs properly.

Purpose of Packaging Comments

When finalizing your wrapped application, you may want to attach a note stating that this is a license-injected package, or provide "next steps" instructions such as how to restart the process with another package, how to restart the License Server, or how to set up usage reporting. The exact text should be decided upon by the development and documentation team.

To associate comments with a package, select the **Modify package comments** option on the [Finalize Package Page](#), and enter comments in the box. These comments are stored in the System Information Stream of the package, and will be visible from Windows Explorer if you right click on this .msi file and select **Properties** from the context menu.

FLEXwrap Assistant Reference

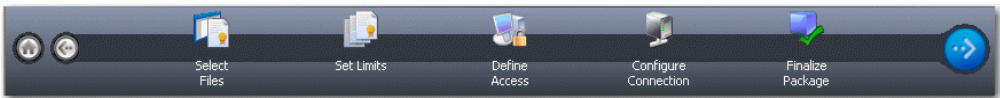
This section describes each of the Pages and dialog boxes that you encounter when using the FLEXwrap Assistant.

- [Home Page](#)
- [Select Files Page](#)
- [Set Limits Page](#)
- [Define Access Page](#)
- [Configure Connection Page](#)
- [Finalize Package Page](#)

- [Permitted Hostid Dialog Box](#)
- [Options Dialog Box](#)
- [Output Window](#)

Home Page

The FLEXwrap Assistant Home Page includes a graphic that helps you visualize the injection process, and includes a Navigation Bar with links to the tasks performed using this Assistant:



The license-injection process is presented on the following five pages:

Table 21-4 • FLEXwrap Assistant Pages

FLEXwrap Assistant Page	Description
Select Files Page	Select a Windows Installer package and the applications within it that you want to license-enable.
Set Limits Page	Specify the number of concurrent licenses available for the license-injected application(s), set the license limits, set the overdraft license limits, and set the queuing behavior.
Define Access Page	Define the handling options for concurrent use of the license-enabled application(s). By default, each application instance (process) requires its own license. You can change this restriction by selecting a different grouping option. You can also restrict application usage by hostid.
Configure Connection Page	Specify a path to an existing server license file, specify how the license-enabled application will find the License Server, how strictly licenses should be enforced , and how to handle loss of connection to the License Server.
Finalize Package Page	Complete the license-injection process by optionally modifying settings and then clicking the Build Package button. All the files in the source Windows Installer package folder will be copied under a subfolder of the output folder you select below, then the license-injection process will start.


To return to this Home Page at any time, click the Home button on the left side of the Navigation Bar:



Select Files Page

On the Select Files page, select a Windows Installer package and the applications within it that you want to license-enable.

Table 21-5 • Select Files Page Options

Option	Description
Select Windows Installer Package	Select a Windows Installer package containing the applications that you want to license-enable. After you select an .msi file, all executables in that package are listed in the Select Executable File(s) to License Inject list.
Select Executable File(s) to License Inject	<p>By default, this list only contains executables that also have associated shortcuts, and all of these executables are selected. This option can be unselected and all executables in the package will display in this list.</p> <p>If multiple executables are selected, the option to treat selected executables as a suite becomes available (by default, all executables with shortcuts are displayed and selected, so this option is by default enabled). If the administrator selects this option, they need to specify a name for the suite they are creating. If you select multiple executables and do not treat them as a suite, then each executable creates its own license file with identical options to other selected executables.</p>
Only show executable files with associated components with shortcuts	By default, only the applications with a shortcut are listed in the Select Executable File(s) to License Inject list. To list all of the executables in the selected Windows Installer package, select this option.
Treat selected executable files as a suite	When a suite is created, all applications in the suite operate under a single license. In this case, a license represents the ability to run each of those executables concurrently. For example, if you had a suite consisting of Word.exe, Excel.exe, and PowerPoint.exe, you would be able to run one instance each of them simultaneously and still only use a single license for the suite. If you select multiple executables and do not treat them as a suite, then each executable creates its own license file with identical options to other selected executables. If you select this option, enter a Suite name to identify the suite.
Encrypt original executable file(s)	<p>Select this option to encrypt the original executable file(s) to prevent users from circumventing license restrictions by running the non-license-injected versions of the executables.</p> <div>  <p>Caution • This encryption feature of FLEXwrap is not guaranteed because some applications can be configured to discourage any alteration, and encryption of the application binary may not be allowed.</p> </div>

Set Limits Page

On the Set Limits page, you specify the number of concurrent licenses available for the license-enabled application, set the license limits, set the overdraft license limits, and set the queuing behavior.


Table 21-6 • Set Limits Page

Option	Description
License limit	Specify the number of concurrent licenses available for the license-enabled application. By default, the limit is set to 1.
Enable overdraft licenses	Select this option if you want to be able to set a number of extra licenses above and beyond the license limit (Overdraft limit). This allows you to cover peak usage times until new licenses can be purchased. By default, this option is not selected.
Overdraft limit	You can optionally specify the number of overdraft licenses available for peak usage.
Set an expiration date for the licenses	Select this option to make the licenses for this package expire on a certain date. By default, this option is not selected.
Licenses expire on	Specify the date on which the licenses will expire. The expiration date defaults to one year from the current date.
Action to take when license limit is reached	<p>Select one of the following options to specify action to take when the license limit has been reached (all licenses are currently in use):</p> <ul style="list-style-type: none">• Ask user—Display a list of current users, and prompt the user attempting to launch the application to choose to wait for a license to become available. The user has the option of contacting one of the other users determine if they plan to close the application soon. If the user queues for a license, then the application will run when a license becomes available. This is the default behavior.• Always queue—Always queue license requests if a license is currently unavailable, and launch the application when a license is available.• Never queue—If a license is not available, inform the user to run the application later.

Define Access Page

On the Define Access page, you can change how license requests are grouped, and you can choose to restrict application usage by computer.

Table 21-7 • Define Access Page Options

Option	Description
Group license requests by	<p>By default, each concurrent request of an application requires one license. However, if you select this option, you can change how license requests are grouped</p> <ul style="list-style-type: none"> • Process—Each request uses one license. (Default) • Host—All concurrent requests from the same computer use only one license, regardless of whether multiple users are accessing the application from the same computer. • User—All concurrent requests from the same user use only one license, regardless of whether the same user is accessing the application from multiple computers. • User + Host—All concurrent requests from the same user on the same computer use only one license. This is the default selection if grouping concurrent license requests. • Site—All concurrent requests from the same site use only one license.
Restrict access by hostids listed below	<p>Select this option to allow access to this license-enabled application only if the host name of the computer is in the Permitted hostids list.</p>
Permitted hostids	<p>List of computers that are permitted access to this license-enabled application. Computer names are listed in the standard Windows format that is displayed on the Computer Name tab of the Microsoft Windows System Properties dialog box.</p> <p>If there are computers in this list and the Restrict access by hostids listed below option is cleared, the list becomes non-editable but is persisted.</p>  <p>Note • To determine a machine's <i>hostid</i>, launch FLEXwrap LMTOOLS on that machine, and view the information on the System Settings tab.</p>
Add	<p>Click to open the Permitted Hostid Dialog Box, where you can enter a computer Hostid Value to add to the Permitted hostids list.</p>
Edit	<p>Select a Permitted hostid and click Edit to open the Permitted Hostid dialog box where you can edit that computer's Hostid Type and Hostid Value.</p>
Delete	<p>Select a Permitted hostid and click Delete to delete it from the Permitted hostids list.</p>


Configure Connection Page

On the Configure Connection page, you specify a path to an existing server license file, enable the application to run if the License Server is unavailable, enable a user to enter the host name and port redefinition at install time, and specify how long the License Server needs to be disconnected before the user is notified.

Table 21-8 • Configure Connection Page Options

Option	Description
FLEXwrap server license (.lic) file	<p>Browse to the license file location. After a license file has been selected, the Host name and Port for the License Server will be displayed.</p> <p>The selected license file will not be modified and a newly generated one will be copied to the same location.</p>
Host name (read only)	The host name of the FLEXwrap License Server listed in the selected license file.
Port (read only)	The port of the FLEXwrap License Server listed in the selected license file.
Allow application to run if the License Server is unavailable	This option stores the license file on the local machine. By default, this is unselected.
Allow host name and port redefinition at install time	In order to accommodate the use case of having multiple servers across different location and needing to specify a different License Server name based on location or any other factor using the same package Select to allow customization of the server location when this application is installed.
Specify timeout value; If license is lost, user is warned and application is closed after this number of minutes	<p>Select this option if you want to specify the number of minutes that an application will continue to run if the License Server becomes unavailable.</p> <p>By default, there is no timeout period, so the application will continue to run without a license and will continue to attempt to regain its license indefinitely.</p> <p>If this option is selected and a Timeout value is entered, the application warns the user and then exits if the license cannot be regained within the timeout period.</p>
Timeout value (minutes)	This option is enabled if the Specify timeout value option is selected. For situations where the License Server becomes unavailable, enter the time in minutes that you want the application to attempt to re-establish a connection to the License Server before the application exits. The minimum amount is 5 minutes.

Table 21-8 • Configure Connection Page Options (cont.)

Option	Description
Request a FLEXwrap server license	<p>Click to open an entry form on the AdminStudio Web site where you request a FLEXwrap evaluation license (eval.lic). The license will be sent to you via email.</p>  <p>Note • The Evaluation License File has a license limit count of 5, which means that only five licenses are available for an application that is FLEX-enabled using this license.</p>

Finalize Package Page

On the Finalize Package page, you specify the location where the wrapped Windows Installer and its source files will be saved and click **Build Package** to create the wrapped application. You can also choose to update the package code or modify package comments.

Table 21-9 • Finalize Package Page

Option	Description
Output folder	<p>Specify the folder where the wrapped Windows Installer application will be saved. All the files in the source Windows Installer package folder will be copied into a subfolder of this output folder.</p> <p>By default, the files will be copied to the path specified on the FLEXwrap Assistant Options dialog box, which is opened by selecting Options on the Tools menu.</p>
Update package code	<p>Select this option to automatically update the package code of this wrapped package to differentiate it from the original package. By default, this option is selected.</p> <p>If the wrapped package used the same package code as the original package, it would cause issues with patches and anything else that would treat both packages the same. If patches intended to update the original application's binary were applied to the wrapped package, the patch would fail. By creating a new package code, this problem is avoided.</p>
Modify package comments	<p>Select this option to note that this is a license-injected package or to enter any other comments that you feel would be helpful.</p> <p>The comments are stored in the MSI Package's System Information Stream and are visible from Windows Explorer if you right click on this .msi file and select Properties from the context menu.</p>
Build Package	Click to create a new Windows Installer package with built-in licensing.




Note • Make sure you restart your licensing server or issue a *reread* command when any license file is modified or added.

Permitted Hostid Dialog Box

On the Permitted Hostid dialog box, you can restrict application usage by hostid, a value that uniquely identifies a machine on a network.

If you enter hostids on this dialog, only users on the specified hostids will be able to use this application.

Table 21-10 • Permitted Hostid Dialog Box Options

Option	Description
Hostid Type	<p>Select the type of hostid that you want to use to restrict application usage. The following types of hostids are supported:</p> <ul style="list-style-type: none"> • Ethernet address • Internet IP address • Machine name • User name • Windows disk serial number • Other
Hostid Value	<p>Value identifying the hostid of the selected type.</p>  <p>Note • You can run <i>lmutil.exe</i> on any machine to get the needed hostid value if you do not know it.</p>

Options Dialog Box

The **Options** dialog box is opened by selecting **Options** on the **Tools** menu and includes the following options

Table 21-11 • Options Dialog Box Options

Option	Description
Default Output Folder	<p>Click Browse and select the default output folder to use when license injecting a package.</p> <p>When using the FLEXwrap Assistant to license-inject a package, the directory you specify in this field will populate the Output folder field on the Finalize Package Page, where you can either accept this default location or specify another location.</p>

Table 21-11 • Options Dialog Box Options

Option	Description
Show confirmation dialogs	To suppress the display of confirmation dialog boxes during the license-injection process, clear this option.

FLEXwrap Server Name Dialog Box

On the FLEXwrap Server Name dialog box, enter the machine name of the computer where the FLEXwrap server is installed. This dialog box opens the first time you open the FLEXwrap evaluation license file (eval.lic) on the [Configure Connection Page](#).

To obtain a FLEXwrap evaluation license file, click the **Request a FLEXwrap server license** link on the [Configure Connection Page](#).



Note • The Evaluation License File has a license limit count of **5**, which means that only five licenses are available for an application that is FLEX-enabled using this license.

Output Window

When you click **Build Package** on the [Finalize Package Page](#) to imitate the creation of a wrapped package, the Output Window opens and displays the messages that are generated during the process, including error messages.

To copy these comments so that you can paste them into another application (such as a word processing program), right-click in the Output window, select **Select All** from the context menu, and then select **Copy** from the context menu.

Managing FLEXwrap License Files

This user documentation explains how to use the FLEXwrap Server Configuration Tool, and is presented in the following sections:

Table 21-12 • FLEXwrap Server Configuration Tool Topics

Topic	Description
Overview of the FLEXwrap License Server	Explains how the FLEXwrap License Server is accessed, and discusses platform and communication issues.

Table 21-12 • FLEXwrap Server Configuration Tool Topics

Topic	Description
Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server	Explains how to install a production FLEXwrap license file, test a wrapped application, start and stop the License Server, check the status of the License Server, and track projects.
Using the FLEXwrap Server Configuration Tool to FLEX-Enable Executable (.exe) Files	Explains how to FLEX-enable an executable file using the FLEXwrap Server Configuration Tool, and explains how to set up failsafe modes for an executable.
FLEXwrap Server Configuration Tool Reference	Documents FLEXwrap terminology and license parameters.



Note • For an overview of AdminStudio Software Licensing technology and the FLEXwrap application, see [About FLEXnet Licensing Technology and FLEXwrap](#).

- To license-enable Windows Installer packages, use the FLEXwrap Assistant. See [Enabling License Tracking of Windows Installer Packages](#) for more information.

Overview of the FLEXwrap License Server

You can manage and monitor your FLEXwrap License Server using the FLEXwrap Server Configuration Tool. The FLEXwrap License Server serves licenses to wrapped applications based on the options that are set and the policies defined in the FLEXwrap license file.



Note • For more information on license files, see [About FLEXwrap License Files](#).

When a user launches a wrapped application, the following occurs:

- **Wrapper launches**—When a user chooses to open an application, the Wrapper launches (which is transparent to the user).
- **Wrapper obtains location of License Server**—The Wrapper obtains the location of the FLEXwrap License Server from the server location file (.1f) or the SWRAP_LICENSE_FILE environment variable.
- **Wrapper contacts License Server to obtain a license**—The Wrapper attempts to obtain a license for the wrapped application from the License Server.
- **Wrapper launches the wrapped application**—If the Wrapper obtains a license for the application, the application is launched.

The following diagram illustrates how the FLEXwrap License Server is accessed:

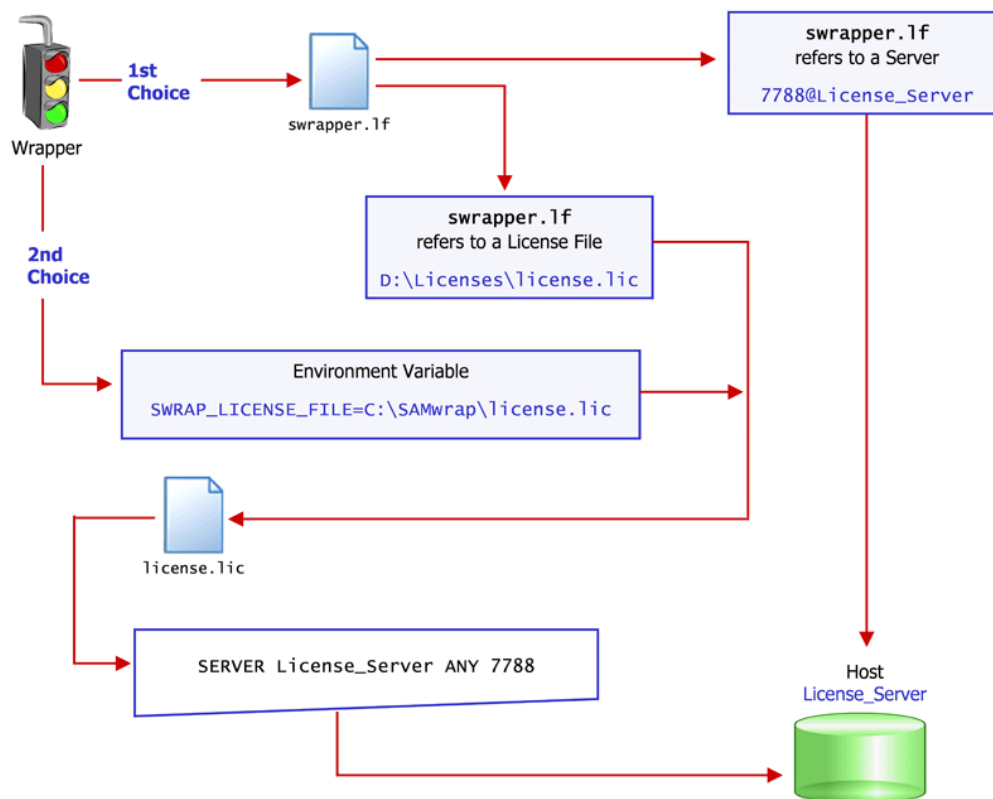


Figure 21-4: Accessing the FLEXwrap License Server

For detailed information on using the FLEXwrap Server Configuration Tool, see [Managing FLEXwrap License Files](#).

Supported Platforms

Applications that run on Microsoft Windows 95 or later can be wrapped, regardless of the platform from which the application is served.

The License Server can run on:

- Windows NT 4 or later
- AIX, HP, Red Hat Linux, SGI, Solaris



Note • For information about FLEX-enabling UNIX applications, contact AdminStudio Support.

Communications Issues

FLEXwrap supports either TCP or UDP communications transport. The default for FLEXwrap is TCP. This should prove satisfactory in all circumstances; however, you have the option to select UDP communications. You can select UDP in either of the following ways:

Table 21-13 • Methods to Select UDP Communications

Method	Description
FLEXLM_COMM_TRANSPORT	Set the FLEXLM_COMM_TRANSPORT environment variable: C:\>set FLEXLM_COMM_TRANSPORT= UDP
COMM TRANSPORT	Select the comm transport in the options file with the line: TRANSPORT UDP

The order of precedence is:

1. Environment variable, if specified
2. Options file entry, if specified
3. Default TCP



Note • When using UDP, FLEXwrap uses a default UDP time out of five minutes.

- Windows systems must be correctly installed and configured with the standard Windows Socket (WinSock) interface.

Installation Notes

When you install the FLEXwrap software, there are some things you should keep in mind:

- **Do not run the License Server as root**—It is not necessary to install or run FLEXwrap as root and it is strongly recommended that the License Server not be run as root. The license file generator (swsetup32.exe and swcrypter.exe), however, should be protected from execution by your general user community.
- **License Server daemons need to be installed locally**—The License Server daemons (lmgrd and swrap) should be local to the system(s) on which they will run. So either install FLEXwrap on the machine where the License Server will run or copy lmgrd and swrap to the machine where the License Server will run. You should not run License Servers on diskless nodes.
- **Binaries to be wrapped need to be able to access the Wrapper**—Each binary to be wrapped must have access to the Wrapper binary, either on a local drive or through a permanent NFS mount point. So either copy the platform-specific Wrapper to each machine that contains a binary to be wrapped or NFS-mount the FLEXwrap drive to all machines that contain binaries to be wrapped.

About the FLEXwrap Server Configuration Tool

You can use the FLEXwrap Server Configuration Tool to monitor the usage of FLEX-enabled applications and manage FLEXwrap license files on the server. The FLEXwrap Server Configuration Tool serves the licenses to wrapped applications and can log usage of the licenses in a FLEXwrap report log. You can then generate usage reports using FLEXnet Manager.

For more information, see [Managing FLEXwrap License Files](#) and [Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server](#).

About the FLEXwrap LMTTOOLS

FLEXwrap LMTTOOLS is a utility used to manage the FLEXwrap License Server. Some of the functions FLEXwrap LMTTOOLS performs include:

- Starting, stopping, and configuring FLEXwrap license server systems.
- Obtaining system information, including hostids.
- Obtaining server status.

The FLEXwrap LMTTOOLS is launched from the Windows **Start** menu.

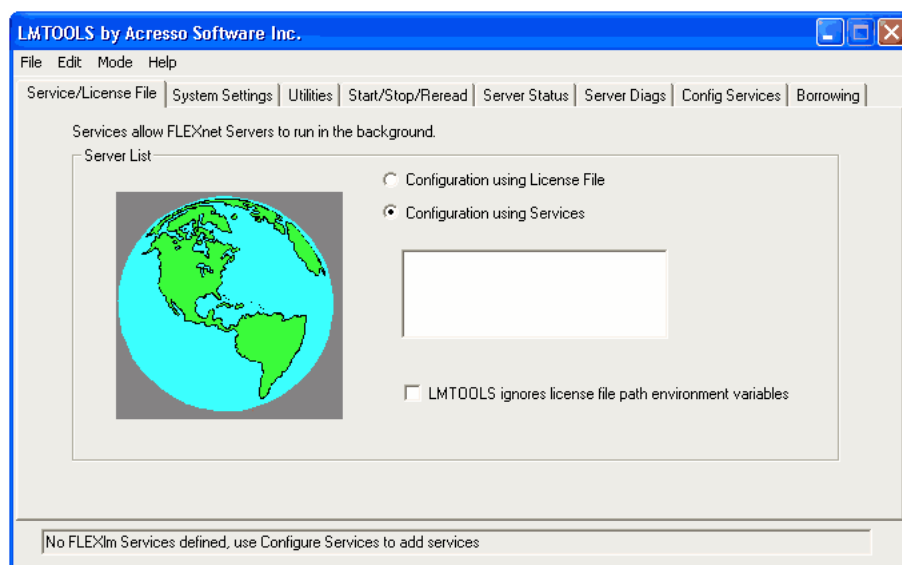


Figure 21-5: FLEXwrap LMTTOOLS interface

LMTTOOLS has two modes in which to configure a license server system:

- [Configuring License Server Using a License File](#)
- [Configuring License Server Using Services](#)

Configuring License Server Using a License File

If you configure your license server system using a license file, operations are performed on a particular license file. The file can be either local or remote. In this mode, you cannot start the `lmgrd` process, but you can do everything else.



Task: *To configure your license server in license file mode:*

1. Launch LMTOOLS.
2. Click the **Configuration using License File** radio button.
3. Enter one or more the license file names or `port@host` specifications.

Configuring License Server Using Services

If you configure your license server system using services, operations are performed on a service, which allows starting `lmgrd` processes local to the system on which LMTOOLS is running. For details on configuring services, see [Configuring FLEXwrap License Server to Run as a Service](#).

Using the FLEXwrap Server Configuration Tool to Manage the FLEXwrap License Server

This section explains how to use the FLEXwrap Server Configuration Tool to manage the FLEXwrap License Server, and includes the following procedures:

- [About the FLEXwrap Server Configuration Tool](#)
- [Installing Your Production FLEXwrap License File](#)
- [Testing a Wrapped Application](#)
- [Starting the License Server](#)
- [Stopping the License Server](#)
- [Checking the Status of the FLEXwrap License Server](#)
- [Tracking Projects with FLEXwrap](#)

Installing Your Production FLEXwrap License File

These instructions assume that you have already installed FLEXwrap and been working with a FLEXwrap evaluation license file (`eval.lic`).



Note • For information on obtaining a FLEXwrap evaluation license file, see [Evaluation License File](#). You can also visit the following site:

<http://www.flexerasoftware.com/flexwrap/eval>

The hostid(s) from the SERVER line(s) of your production FLEXwrap license file is encrypted into the license keys generated for FEATURE lines for wrapped applications that you add to the FLEXwrap license. You can update your FLEXwrap evaluation license file with the production FLEXwrap license information either using the FLEXwrap Server Configuration Tool or manually:

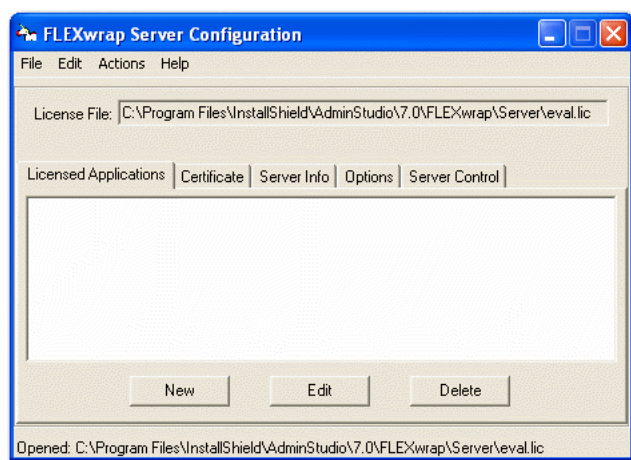
- **Using the FLEXwrap Server Configuration Tool**—If you update your FLEXwrap evaluation license with your production FLEXwrap license information using the FLEXwrap Server Configuration Tool, any license keys that you generated using your evaluation license will be regenerated automatically.
- **Manually**—If you manually update your FLEXwrap evaluation license file with your production FLEXwrap license information by editing the file, you will have to use the FLEXwrap License Encryption Utility (swcrypter.exe) to regenerate its license keys for wrapped applications. The FLEXwrap License Encryption Utility is used to manually sign FLEXwrap licenses.



Task:

To update your FLEXwrap license file from the evaluation file:

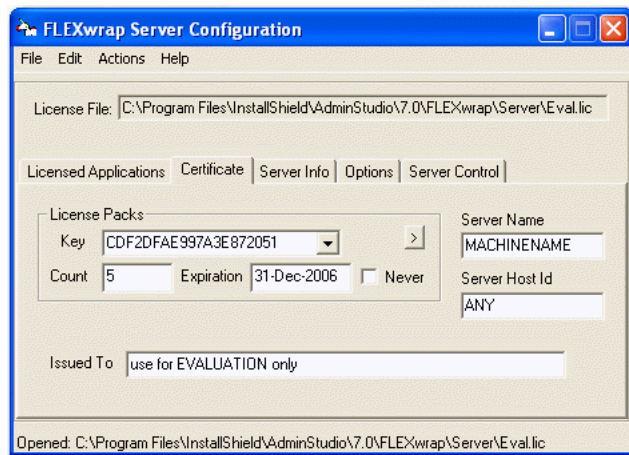
1. Backup your FLEXwrap evaluation license file (eval.lic) to another file name (such as eval1.lic) so that you do not accidentally modify it. Using a different extension than .lic is recommended to prevent the Server from automatically loading the license file when it starts.
2. Launch the FLEXwrap Server Configuration Tool from the Windows **Start** menu. The **FLEXwrap Server Configuration** dialog box opens.



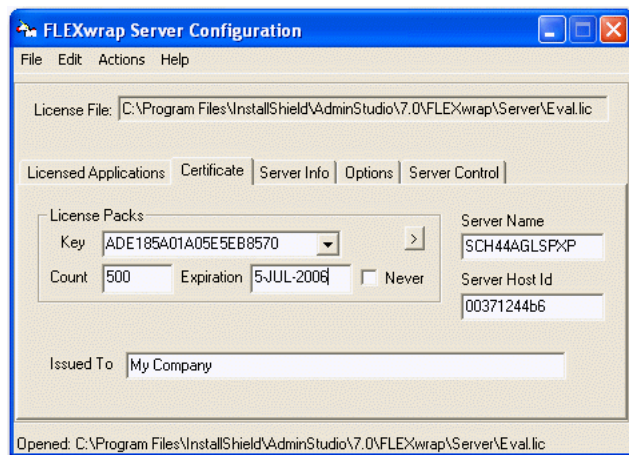
You use this dialog box to install your FLEXwrap license file, to generate licenses for the applications to be wrapped, and to wrap individual executable files.

3. If you have not already opened your evaluation license file, click **Open** on the **File** menu and select your FLEXwrap evaluation license file (eval.lic).

4. If you are running the Server, open the **Server Control Tab** and stop the Server.
5. Click the **Certificate** tab. The **Certificate Tab** opens.



6. Replace the entries in the **Certificate Tab** fields with the exact production FLEXwrap license information you received from AdminStudio Support.
 - **Key**—License key provided by AdminStudio Support.
 - **Count**—The number of concurrent licenses available for this FLEX-enabled application.
 - **Expiration**—The date these licenses will expire.
 - **Server Host Id**—The host name of the FLEXwrap License Server listed in the production license file.
 - **Issued To**—The name of the company that purchased the license.
7. Enter the **Server Name** associated with the **Server Host Id** that you entered.



8. On the **File** menu, click **Save** to save the license file and enter a name using the .lic extension, such as flexwrap.lic.

Because you have updated your FLEXwrap license using the FLEXwrap Server Configuration Tool, any FEATURE lines in the license file that you created with your FLEXwrap evaluation license information will be rekeyed using your production FLEXwrap license information.

9. Start the FLEXwrap License Server from the **Server Control** tab.



Note • If you generate a new Windows Installer package using the FLEXwrap Assistant using the evaluation license file, and if you had selected the option to run even if the server is down, you should rewrap the package once you receive the production license because the local license copy installed will still be the old one.

Starting the License Server

There are three options for starting the FLEXwrap License Server:

Table 21-14 • Methods of Launching the FLEXwrap License Server

Option	Description
Configuring FLEXwrap License Server to Run as a Service	Launch an installation from the FLEXwrap interface to configure the License Server to run as a service (to start automatically when the system is booted).
Manually Starting the FLEXwrap License Server from the FLEXwrap Interface	Click a button in the FLEXwrap interface.
Manually Starting the FLEXwrap License Server from the Command Line	Run a command at a Windows console prompt.



Note • When you purchase FLEXwrap, you specify the *hostid* to which the License Server will be locked.

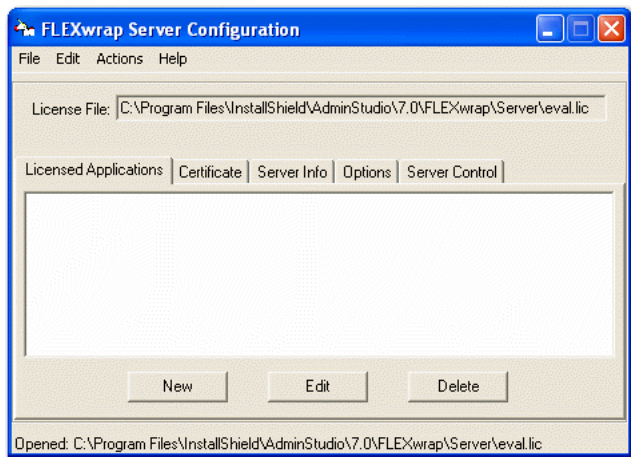
Configuring FLEXwrap License Server to Run as a Service

You can configure the FLEXwrap License Server to run as a service by selecting the **NT Services** option on the **Server Control** tab of the **FLEXwrap Server Configuration** dialog box.

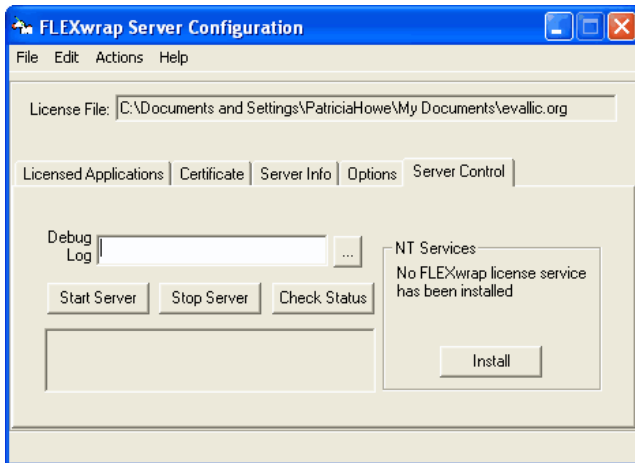


Task: *To configure the FLEXwrap License Server to run as a service:*

1. Launch the FLEXwrap Server Configuration Tool from the **Start** menu. The **FLEXwrap Server Configuration** dialog box opens.



2. On the **File** menu, click **Open** and select the FLEXwrap license file.
3. Click the **Server Control** tab. The **Server Control** tab opens.



4. Click the **Install** button in the **NT Services** area. The selected license file is now configured to run as a service.

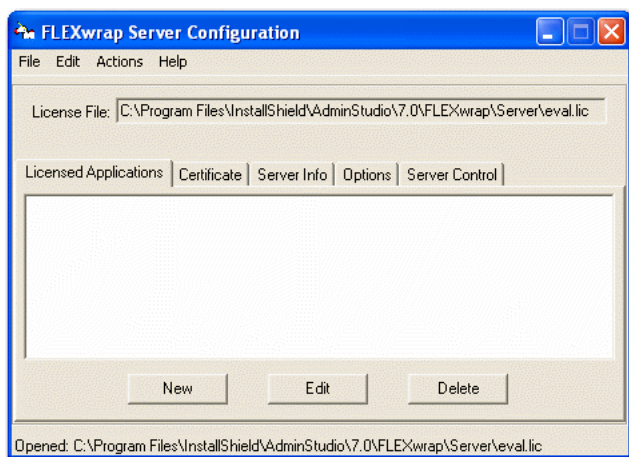
Manually Starting the FLEXwrap License Server from the FLEXwrap Interface

You can manually start the FLEXwrap License Server on the **Server Control** tab of the **FLEXwrap Server Configuration** dialog box.

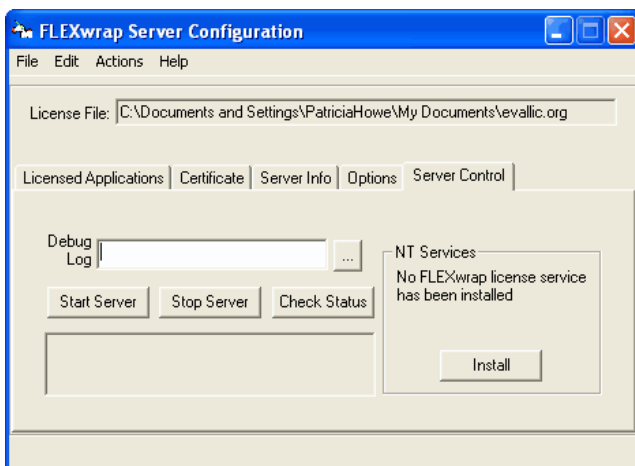


Task: *To start the FLEXwrap License Server manually from the FLEXwrap interface:*

1. Launch the FLEXwrap Server Configuration Tool from the **Start** menu. The **FLEXwrap Server Configuration** dialog box opens:



2. On the **File** menu, click **Open** and select the FLEXwrap license file.
3. Click the **Server Control** tab. The **Server Control Tab** opens.



4. In the **Debug Log** field, enter the path to a debug log or select an existing debug log.



Note • If you do not specify a debug log when you start the License Server, at least two console windows will appear—one for the `lmgrd` output, one for the `swrap` output, and one for each of any other vendor daemons started by this `lmgrd`.

5. Click the **Start Server** button. All licenses (*.lic) in the directory of the opened license are loaded.

Manually Starting the FLEXwrap License Server from the Command Line

You can manually start the FLEXwrap License Server by running a command at the Windows console prompt.



Task: *To start the FLEXwrap License Server from the command line:*

Run the following command at a Windows console prompt:

`C:\path_to_lmgrd.exe -c path_to_license_file`

For example:

`"C:\Program Files\SAMwrap\v8.2\lmgrd.exe" -c "C:\Program Files\SAMwrap\v8.2\FLEXwrap.lic"`

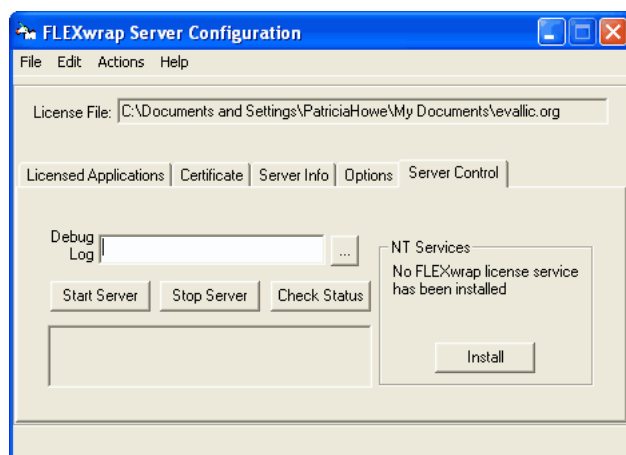
Stopping the License Server

You can stop the FLEXwrap License Server on the **Server Control** tab of the **FLEXwrap Server Configuration** dialog box.



Task: *To stop the License Server:*

1. Launch the FLEXwrap Server Configuration Tool from the **Start** menu. The **Licensed Applications** tab of the **FLEXwrap Server Configuration** dialog box opens.
2. On the **File** menu, click **Open** and select the FLEXwrap license file.
3. Click the **Server Control** tab. The **Server Control** tab opens.



4. Click **Stop Server**.

Checking the Status of the FLEXwrap License Server

You can check the status of the FLEXwrap License Server on either the **Server Control** tab of the **FLEXwrap Server Configuration** dialog box or on the **Server Status** tab of the **LMTOOLS** dialog box. The LMTOOLS status report is slightly more detailed than the report generated by the FLEXnet interface.

Checking the License Server Status from the FLEXwrap Server Configuration Tool

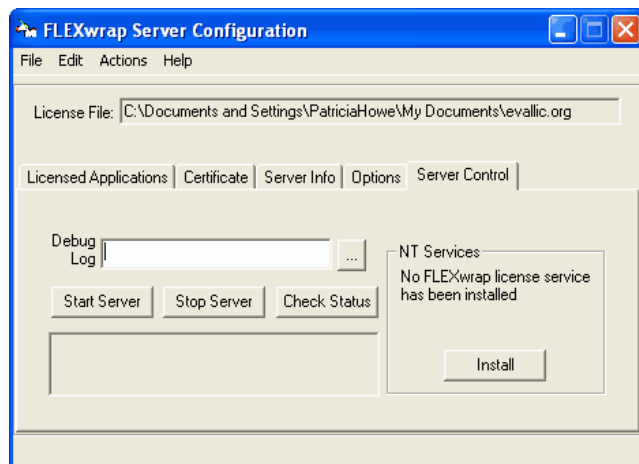
You can check the status of the FLEXwrap License Server using the FLEXwrap Server Configuration Tool.



Task: *To check the status of the License Server from the FLEXwrap interface:*

1. Launch the FLEXwrap Server Configuration Tool from the **Start** menu. The **Licensed Applications** tab of the **FLEXwrap Configuration** dialog box opens.
2. On the **File** menu, click **Open** and select the FLEXwrap license file.

3. Click the **Server Control** tab. The **Server Control** tab opens.



4. Click **Check Status**. A status message is displayed.

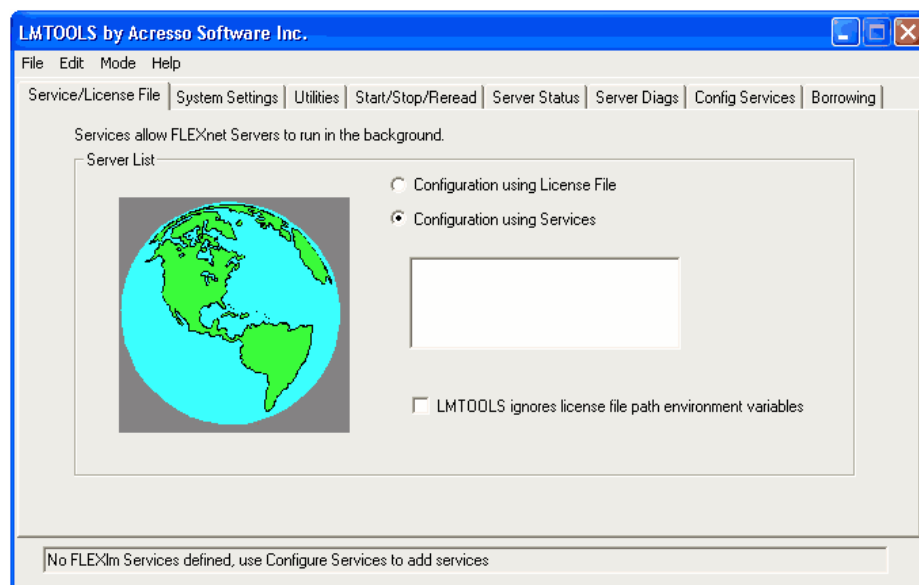
Checking the License Server Status from the LMTTOOLS Interface

You can use the FLEXwrap LMTTOOLS utility to check the status of the FLEXwrap License Server.



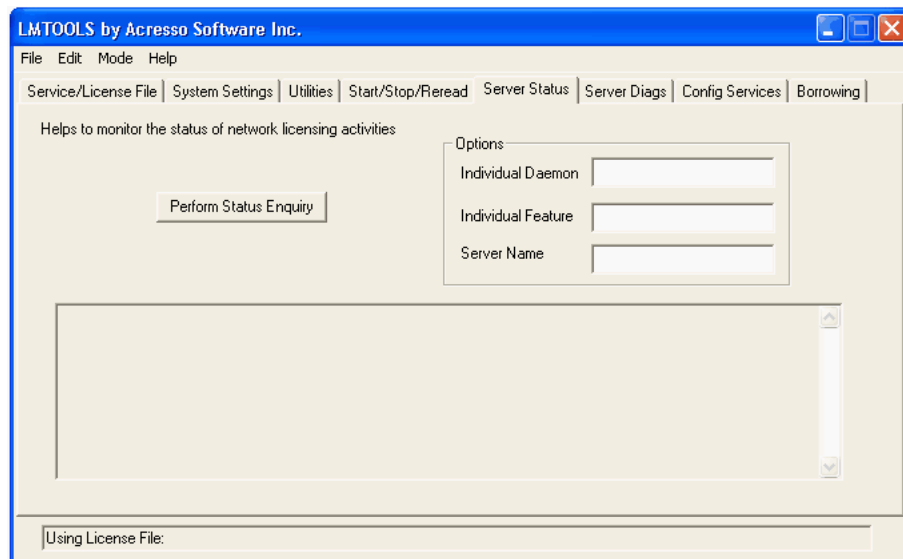
Task: *To check the status of the License Server from the LMTTOOLS interface:*

1. Open the **FLEXwrap LMTTOOLS** utility from the **Start** menu. The **Service/License File** tab of the **LMTTOOLS** dialog box opens.



2. On the **File** menu, click **Open** and select the FLEXwrap license file.

3. Click the **Server Status** tab. The Server Status tab opens:



4. Click **Perform Status Inquiry**. A Status Report opens.

Tracking Projects with FLEXwrap

FLEXwrap allows you to track usage of FLEXwrap-managed applications by project (or department or account). Prior to running the wrapped application, the user must set the LM_PROJECT environment variable to the name of the project to be tracked. Enter the following at the command prompt:

```
C:\> set LM_PROJECT=my_project
```

The project name is written to the report log when a license checkout occurs. You can then filter, summarize, and sort on these projects when you generate reports using FLEXnet Manager.

Using the FLEXwrap Server Configuration Tool to FLEX-Enable Executable (.exe) Files

In addition to managing the FLEXwrap Server, you can also use the FLEXwrap Server Configuration Tool to FLEX-enable an executable (.exe) file. The procedure for doing so are in this section.

However, if you want to FLEX-enable all of the executables in a Windows Installer package, use the FLEXwrap Assistant. See [Enabling License Tracking of Windows Installer Packages](#).

Overview

Putting an application under FLEXwrap management involves the following five basic steps:

Table 21-15 • Basic Steps of Wrapping an Application

#	Step Name	Description
1	Add FEATURE Line to License File	Generate a FEATURE line for the application to be wrapped and add this FEATURE line to the FLEXwrap license file.
2	Start the FLEXwrap License Server	Make the FLEXwrap License Server serve licenses for the new feature by issuing a “reread” command (if the License Server is already running) or by starting the FLEXwrap License Server (if it is not already running).
3	Rename Executable File	Rename the application executable to use a .fwr extension.
4	Replace Original Executable File with Wrapper File	Copy the Wrapper file (swrap_nt.exe) to the application’s location, and give it the original name of the application.
5	Identify License File Path	In the wrapped application’s directory, create a license file named swrapper.1f, a plain text file containing port@host for the 1mgrd process (a License Server daemon) and/or the path to a copy of the license file. swrapper.1f is used by the Wrapper binary to request a license from the License Server.

These basic steps for wrapping an application are illustrated in the following diagram:

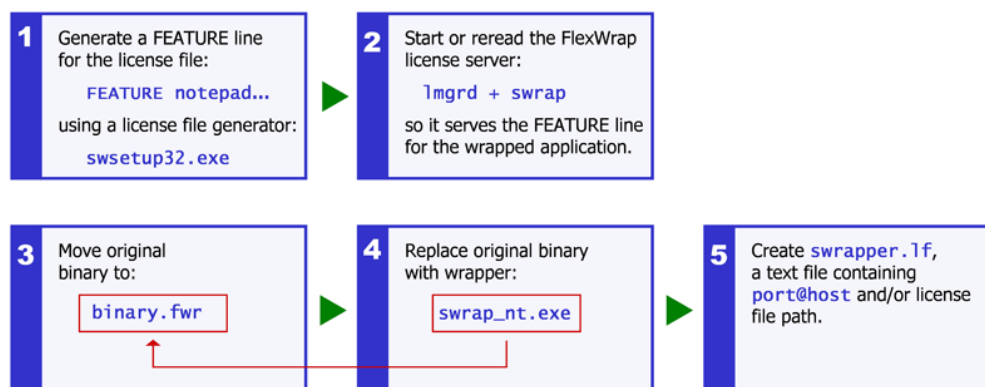


Figure 21-6: Basic steps for wrapping an application

When following these steps to wrap an application, note the following:

- If an application is installed locally on multiple machines, **Step 3** through **Step 5** must be repeated for each installed copy.

- For License Server failsafe mode, additional steps are needed. See [Setting Up License Server Failsafe Mode](#) for instructions.

After you perform these steps, no additional setup needs be done by your end users. The following diagram provides an overview of the basic steps for wrapping an executable file:

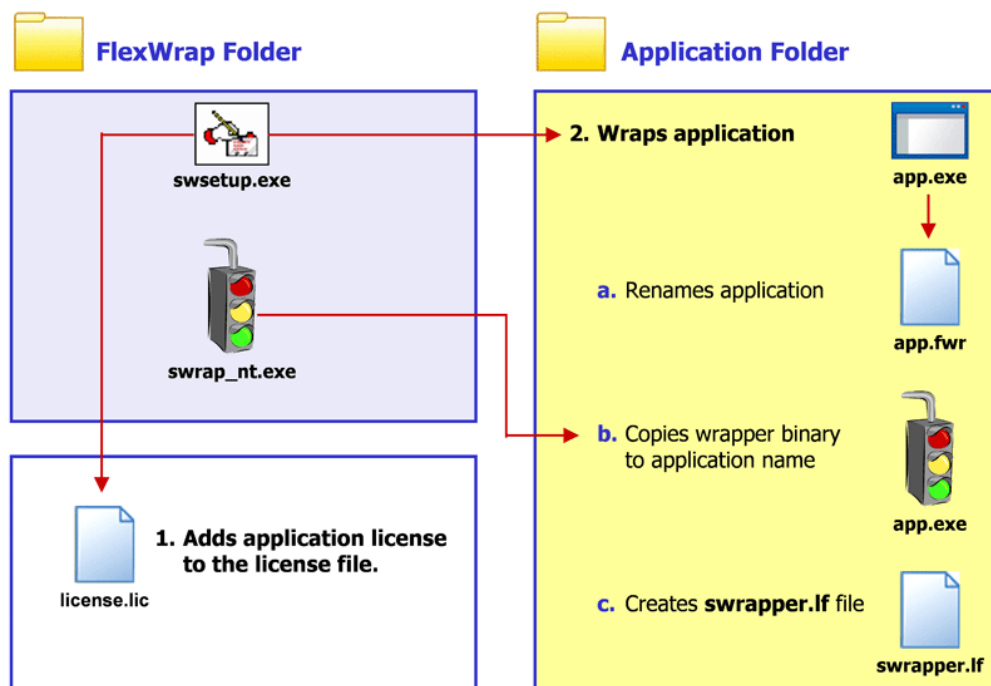


Figure 21-7: Saving the license file and application



Note • To license-enable Windows Installer packages, use the FLEXwrap Assistant. See [Enabling License Tracking of Windows Installer Packages](#) for more information.

Available Options

When using the FLEXwrap Server Configuration Tool to wrap an executable, you can set the same limits, behavior, restriction, and configuration options that you can set when using the FLEXwrap Assistant to wrap a Windows Installer package.

Table 21-16 • FLEXwrap Server Configuration Tool Options when Wrapping an Executable

Option	Description
Set License Limit	Restrict usage of a software application to a specified number of users to run on any machine (counted).
Restrict Usage by Machine	Restrict usage of a software application to either an unlimited number of users or a specified number of users on a specified computer
Expiration Date	Set an expiration date.
Overdraft Support	Allow overdraft usage (reports show usage in excess of license limit).
Queuing	Permit queueing when licenses are not available.
Suite Option	Enforce licensing terms and conditions of product suites.
Restrict Usage by Hostids	Reserve licenses for particular users, machines, or sites.
Group License Requests	Include or exclude specific users, hosts, or displays or groups of users or hosts.
Reporting	Logging license usage

FLEX-Enabling an Executable (.exe) File

License-enabling an executable (.exe) file is a four step process:

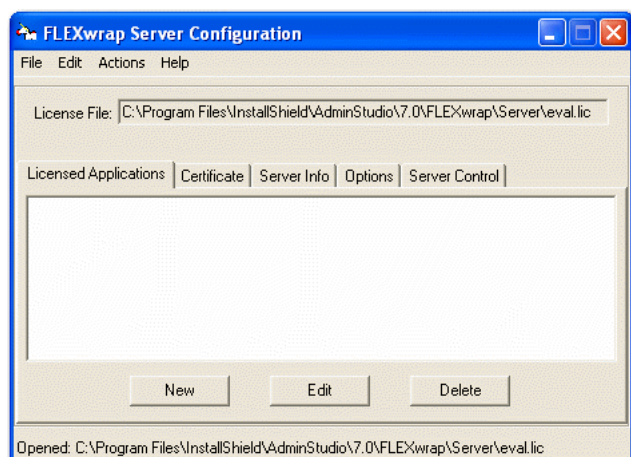
- Step 1: [Specifying Basic Settings for the License](#)
- Step 2: [Specifying Advanced Settings for the License and Binary](#)
- Step 3: [Creating an Options File](#)
- Step 4: [Saving the New License File and Wrapped Application](#)

Specifying Basic Settings for the License

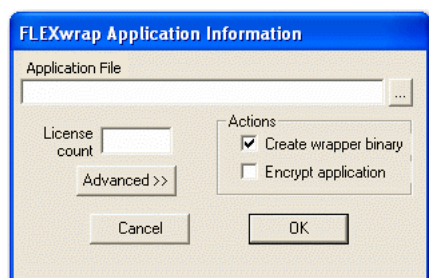


Task: *To specify basic settings for the license:*

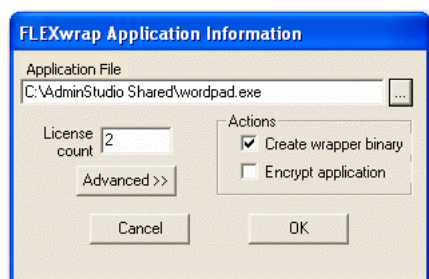
1. Launch the FLEXwrap Server Configuration Tool from the Windows **Start** menu. You will be prompted whether to open the last license file that you worked with in the FLEXwrap Setup dialog box.
2. Click the **No** button. The **FLEXwrap Server Configuration** dialog box opens.



3. On the **File** menu, click **Open** and select your production FLEXwrap license file. After you select the license file, the **FLEXwrap Server Configuration** dialog box displays the path to this license file and a list of programs (if any) that have been licensed with FLEXwrap.
4. Click **New**. The **FLEXwrap Application Information** dialog box opens.



5. In the **Application File** field, select the application binary that you are going to wrap.
6. In the **License Count** field, type the license limit for the application. This is the number of licenses that the License Server will allow to be checked out simultaneously for this application. The license limit will appear in reports as the maximum number of licenses available. The total of the counts of licenses on all the FEATURE lines you generate may not exceed the count on the FLEXwrap INCREMENT line in your license file. If you are just testing FLEXwrap's functionality, enter 2.
7. Make sure the **Create Wrapper binary** option is selected.



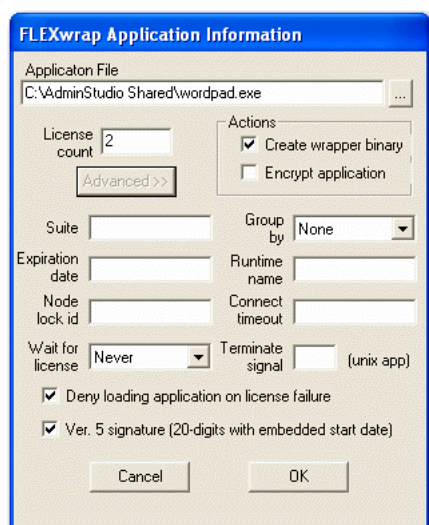
8. Continue with the steps in [Specifying Advanced Settings for the License and Binary](#).

Specifying Advanced Settings for the License and Binary



Task: *To specify advanced settings for the license and binary:*

1. On the **FLEXwrap Application Information** dialog box, click the **Advanced** button. The dialog box expands and the advanced licensing options appear.



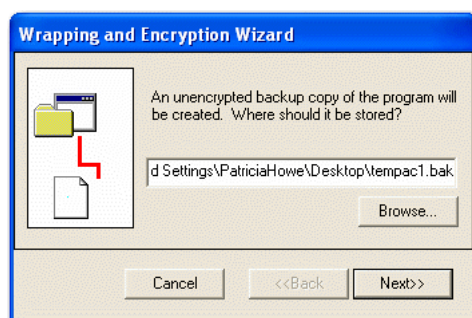
2. Specify the desired options. See [Specifying FLEXwrap License Parameters](#) for detailed instructions.

The most important advanced option to consider is whether you want the wrapped application to be able to run without being able to check out a license from a running License Server (License Failsafe Mode). By default, the **Deny loading application on license failure** option is selected.

- **Run when no license is available**—If you want the wrapped application to be able to run whether or not it can get a license from a running FLEXwrap License Server (License Failsafe Mode), clear the **Deny loading application on license failure** option. See [Setting Up Regular \(License\) Failsafe Mode](#).
- **Run when cannot connect to License Server**—If you want the wrapped application to be able to run whether or not the FLEXwrap License Server is running (License Server Failsafe Mode), clear the **Deny**

loading application on license failure option and follow the instructions in [Setting Up License Server Failsafe Mode](#).

- **Cannot run without a license**—If you wish to require the Wrapper to obtain a license in order to run the wrapped application, select the **Deny loading application on license failure** option.
3. In some cases, the application binary can be encrypted so that it can be run only by the Wrapper. This would prevent users from running the wrapped application binary (binary.fwr) directly. If you want to encrypt the application binary, select the **Encrypt application** option.
 4. Click **OK**. If you selected the **Encrypt application** check box, the **Wrapping and Encryption Wizard** dialog box opens in which you can specify where you would like a backup copy of the application binary to be saved. The default is binary.bak in the application directory.



If you did not select the **Encrypt application** option skip to **Step 11**.

5. Click **Next**. The second panel of the **Wrapping and Encryption Wizard** opens, where you are prompted to identify the path to the License Server.



6. Select the **Host and port number reference** option and enter the host and port number, or select the **Path to the license file** option and enter the appropriate path. This information is written into swrapper.1f. You would normally want to select port@host.
7. Click **Next**. You are prompted to identify the location where you want FLEXwrap will save a copy of the original binary prior to encryption.



8. Identify the location and click **Next**. A message appears stating that FLEXwrap is ready to encrypt and wrap the application.
9. Click **Finish**. A message appears confirming that the application was encrypted successfully.
10. Click **OK**. You have successfully wrapped and encrypted the executable. Now continue with [Creating an Options File](#).

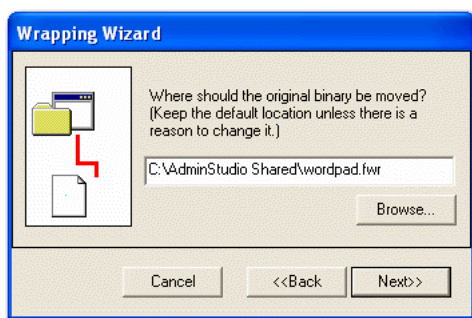


Note • This encryption feature of FLEXwrap is not guaranteed because some applications can be configured to discourage any alteration, and encryption of the application binary may not be allowed.

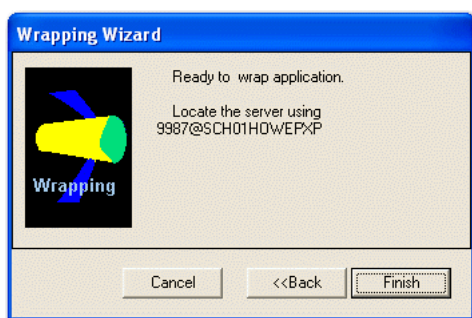
11. If you did not choose to encrypt the application, click **OK** in the **FLEXwrap Application Information** dialog box. The **Wrapping Wizard** dialog box opens and you are prompted to select how you want the Wrapper to find the FLEXwrap License Server.



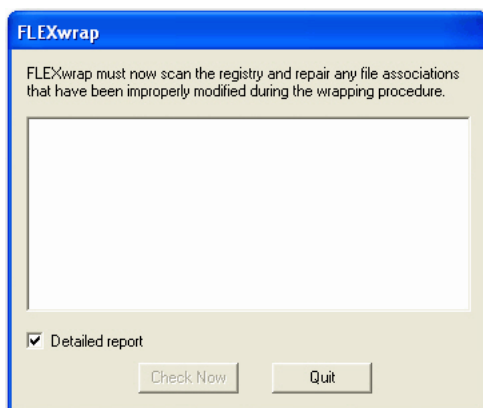
12. Select the **Host and port number reference** option and enter the host and port number, or select the **Path to the license file** option and enter the appropriate path. This information is written into swrapper.1f. You would normally want to select **port@host**.
13. Click **Next**. If you are setting up an uncounted node-locked license, you will be prompted to select whether you want uncounted node-locked licenses to be logged. If you select both logging uncounted node-locked licenses and failsafe mode, failsafe mode may not work correctly.
14. Decide whether to select this option and click **Next**. You are prompted to specify where you would like a backup copy of the application binary to be saved. The default is binary.fwr in the application directory.



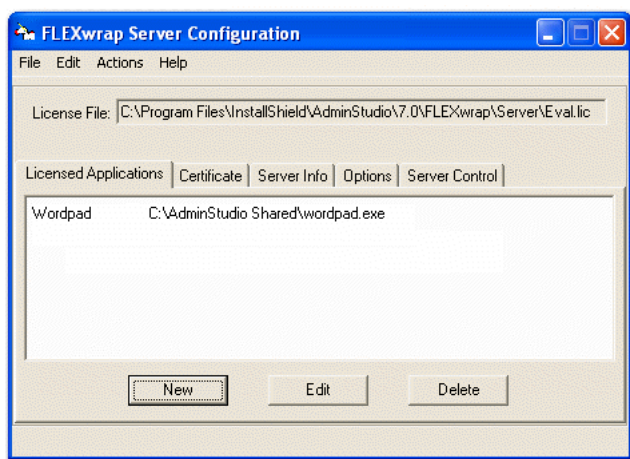
15. Specify the location and click **Next**. A message appears stating that FLEXwrap is ready to wrap the application.



16. Click **Finish**. A command window and a FLEXwrap dialog box appear.



17. Click **Check Now** to scan/repair the registry.
18. When the dialog box displays **Done**, click **Quit**. The **Wrapping Wizard** dialog box should display **Finished**. The FLEXwrap license has been modified, but not yet saved.
19. Click **OK** in the **Wrapping Wizard** dialog box. In the **FLEXwrap Server Configuration** dialog box, you should now see the application in the list of **Licensed Applications**.



20. Continue with the steps in [Creating an Options File](#).

Creating an Options File

If you want to record license usage in a report log, create a FLEXwrap options file. You could then generate usage reports using FLEXnet Manager.



Task:

To create an options file:

1. Create a plain text file called `swrap.opt` and save it in the same directory with your license file.
2. Type a `REPORTLOG` line in the options file:

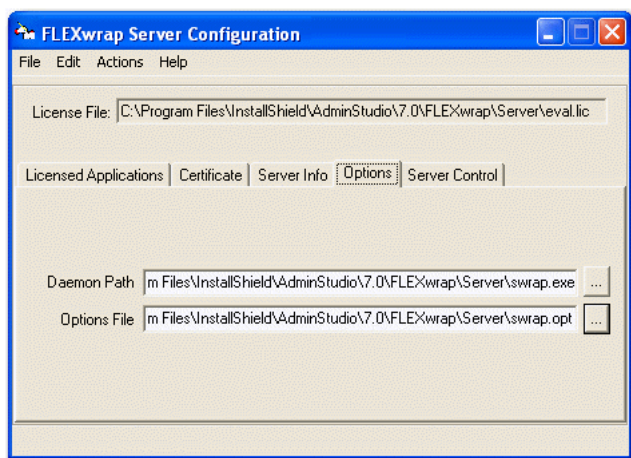
```
REPORTLOG +path_to_report_log
```

For example:

```
REPORTLOG +C:\flexlm\swrap\swrap.rl
```

If you already have a FLEXwrap options file, edit it to contain a `REPORTLOG` line.

3. In the **FLEXwrap Server Configuration** dialog box, click the **Options** tab.
4. In the **Options File** box, enter the path (including file name) to the FLEXwrap options file.



5. Continue with the steps in [Saving the New License File and Wrapped Application](#).

Setting the Path to the FLEXwrap Options File

When you save the FLEXwrap license file in the next section, the path to the FLEXwrap options file will follow the path to the swrap daemon:

```
DAEMON swrap swrap.exe path_to_options_file
```

For example:

```
DAEMON swrap swrap.exe C:\flex1m\swrap\swrap.opt
```

When 1mgrd starts the swrap daemon, the daemon will read the options file and append license usage information to the report log.

Saving the New License File and Wrapped Application



Task: *To save the new license file and wrapped application:*

1. On the **File** menu of the **FLEXwrap Server Configuration** dialog box, click **Save** to save the updated FLEXwrap license file in its original location.
2. In the folder that contains the application binary file, you will see three files:
 - program.exe is now a copy of the Wrapper binary (notice the difference in file size)
 - program.fwr is the original program file that has been renamed
 - swrapper.1f is the location file that the Wrapper binary uses to locate the License Server

Testing a Wrapped Application

When you launch the wrapped application, the Wrapper reads the location file (`swrapper.1f`) to obtain the location of the License Server (`port@host`), contacts the License Server at that location, obtains a license, and launches the wrapped application (`binary.fwr`).



Task: *To test the wrapped application:*

1. Launch the wrapped application in the same way as you launched it before it was wrapped, such as double-clicking the icon.



Note • If the application doesn't run, and you chose to encrypt the application, it may be that the application cannot be encrypted. Restore the backup copy of the application binary and rewrap the binary without encrypting.

2. Check the status of the License Server either with LMTOOLS or at the command line with the following command:

```
% lmutil lmstat -a -c path_to_license_file
```

3. To test what happens when all available licenses are checked out, invoke the wrapped application up to its license limit and then invoke one more.
 - **If you have set up license failsafe mode**, the additional instance of the wrapped application should run.
 - **If you did not set up license failsafe mode**, the additional instance of the wrapped application should not run when it fails to get a license. Instead, you will see a message like the following if you set the Wait for license choice list to **Ask** or **Always**:

The maximum number of users for "application" has been reached.

Do you wish to be notified when a license becomes available?

Current user(s) are:

dodi at zippy on /dev/ttya (v1.0), started on Tuesday 6/17 at 17:51

dodi at speedy on /dev/ttyq8 (v1.0), started on Tuesday 6/17 at 17:51

If you choose Yes, the license server will queue for a license. You will see a dialog that says, "Waiting for [application] license" and when a license becomes available, the application will run.

4. To generate a report of your usage, invoke the wrapped application a few more times, allowing some time to elapse before checking licenses back in.
5. Then send a reread command to the License Server. The reason for the reread is that v5+ report logs are compressed and authenticated and information is written to the report log only periodically. A reread flushes

all usage information into the report log. If you require up-to-the-minute reporting, you must send a reread command to the License Server before generating a report.

A reread can be done using the LMTOOLS program or at the command line with the following command:

```
C:\Program Files\SAMwrap\lmutil lmreread -c C:\Program Files\SAMwrap\v8.2\FLEXwrap.lic
```

Now you can generate a license usage report using [FLEXnet Manager](#).

Setting Up Fail Safe Modes

There are two instances when a wrapped application will be unable to obtain a license. You may want to allow a wrapped application to run without a license in either or both of these instances:

- **Regular (License) Fail Safe Mode**—The Wrapper contacts the License Server, but all licenses for the feature are checked out. Having failsafe mode specified in the license file allows the wrapped application to run without a license. This mode relies on the Wrapper contacting the License Server.
- **License Server Fail Safe Mode**—The Wrapper is unable to contact the License Server, either because the License Server is not running or because of a network problem. Having failsafe mode specified in the license file and ensuring that the Wrapper can read a current local copy of the license file allows the wrapped application to run without a license.

The topics in this section explain how to set up both license and License Server failsafe modes:

- [Setting Up Regular \(License\) Failsafe Mode](#)
- [Setting Up License Server Failsafe Mode](#)

Setting Up Regular (License) Failsafe Mode

In the case where the Wrapper can contact the License Server but all licenses for the feature are checked out, a specification of failsafe mode in the FEATURE line for the wrapped application allows the wrapped application to run without a license.

The Wrapper uses port@host in swrapper.1f to contact the License Server and learns from the License Server that, although all licenses are checked out, failsafe mode was enabled in the license file. A denial is logged in the report log, the Wrapper runs the application binary, and the usage of the wrapped application is not recorded.

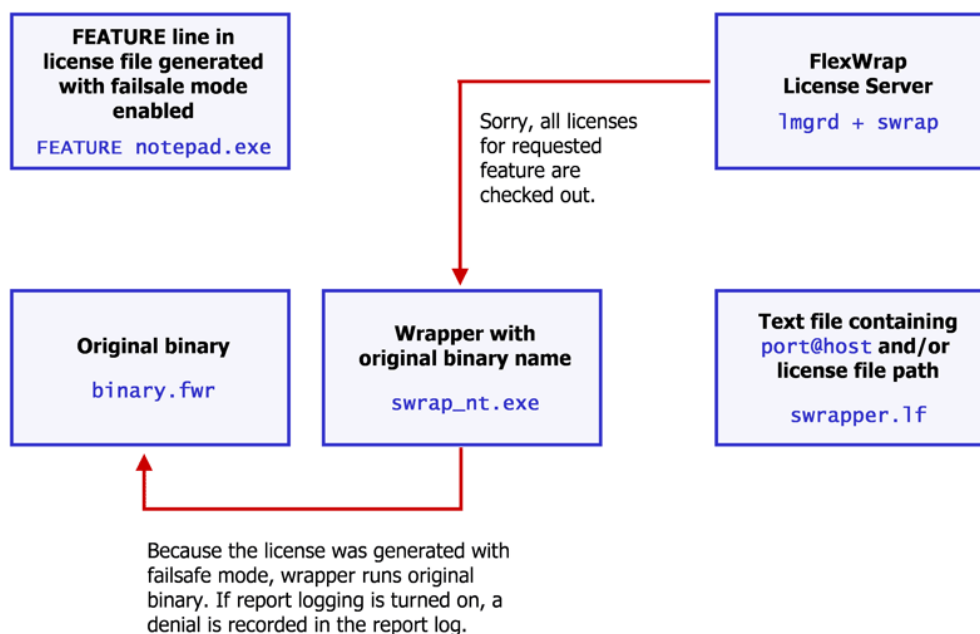


Figure 21-8: Regular (License) Failsafe Mode

This level of failsafe is often adequate. For example, if the wrapped application is being served from the same machine on which the FLEXwrap License Server runs, network communication is not a concern—keeping the License Server up and running would be the only thing necessary to ensure that the wrapped application can always run.

Setting Up License Server Failsafe Mode

In the case where the Wrapper cannot contact the License Server, if port@host is the only information in swrapper.lf, there is no way for the Wrapper to learn that failsafe mode was enabled in the license file.

Unless steps have been taken to ensure that the Wrapper can read a copy of the license file, the wrapped application will not run, even though failsafe mode was specified in the license file. The drawback to ensuring access to the license file is that you may need to make and maintain extra copies of the license file.

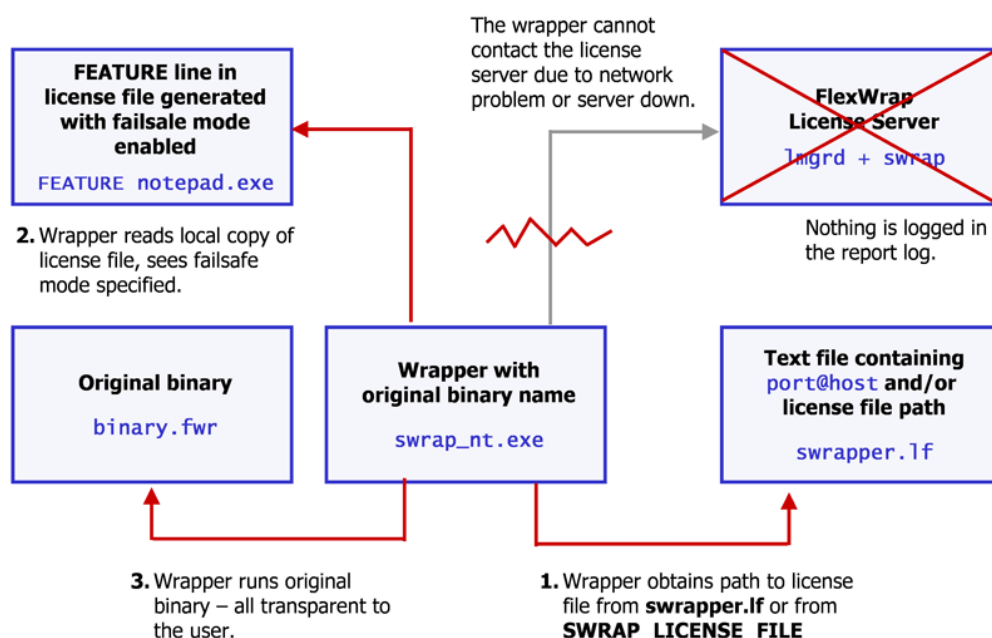


Figure 21-9: License Server Failsafe Mode

Perform the following steps to ensure License Server failsafe.



Task: *To ensure License Server failsafe:*

1. Make sure you specified failsafe mode when you generated the FEATURE line for the wrapped application. The FEATURE line will contain the words: `VENDOR_STRING=failsafe`. This is part of the license key, so you cannot just add these words to an existing FEATURE line; you must specify failsafe mode when generating the license. See [FLEX-Enabling an Executable \(.exe\) File](#) for instructions on generating license files.
2. If the FLEXwrap License Server runs on a different machine from the machine where the wrapped binary resides, remove the network dependency by copying the FLEXwrap license file to a drive local to the wrapped binary. We recommend putting the license file in the directory where the wrapped binary resides and calling it `FLEXwrap.1ic`.
3. Edit `swrapper.1f` in the directory where the wrapped binary resides and add a semi-colon (;) and the path to the license file on the local disk. For example:

```
7788@machinea;"C:\Program Files\SAMwrap\v8.2\FLEXwrap.1ic
```

If, when installing FLEXwrap on Windows, you chose to use the path to the license file instead of `port@host` to find the FLEXwrap license file, you will not have to edit `swrapper.1f` because it already contains the information it needs to find the license file.

The Wrapper will also check the `SWRAP_LICENSE_FILE` environment variable after checking `swrapper.1f`.



Note • FLEXwrap does not replace a built-in license manager. If you wrap an application that has a built-in license manager, the application must obtain a license from FLEXwrap in order to try to ask for a license from its built-in license manager. By enabling failsafe mode (such as allowing a wrapped application to run even if it cannot get a license from a running license server), you can guarantee that a FLEXwrap License Server will never prevent the application from getting to its own license manager.

FLEXwrap Server Configuration Tool Reference

FLEXwrap Reference information is organized into the following areas:

- [FLEXwrap Terminology](#)
- [About FLEXwrap License Parameters](#)

FLEXwrap Terminology

The following terms are defined in the context of the FLEXwrap software and are used throughout this documentation:

Table 21-17 • FLEXwrap Terminology

Term	Description
Application	In this documentation, an application consists of one binary file to be put under FLEXwrap license management. The application might be a word processor, compiler, or CAD tool, for example.
Failsafe mode or License failsafe mode	A mode of operation that allows the Wrapper to run the wrapped application without a license when the FLEXwrap License Server is running. This mode is enabled by specifying failsafe mode when the license is generated. See Setting Up Fail Safe Modes for more information.
FEATURE line	A line of text generated by the FLEXwrap administrator at your site, using a license file generator. This line describes the license policies for a wrapped application and is appended to your FLEXwrap license file.
License	A software “token” that represents the right to use a wrapped application.
License administration tools	Programs that allow the FLEXwrap administrator to manage the License Server: FLEX [™] utilities— <code>lmutil.exe</code> , <code>lmtools.exe</code> (LMT00LS)

Table 21-17 • FLEXwrap Terminology (cont.)

Term	Description
License file	A text file containing the FEATURE lines for all the applications under FLEXwrap management. This file also contains information about where the License Server runs, the location of the swrap daemon, and the location of the options file. The license file is usually called FLEXwrap.lic or license.dat and is read by the License Server and, under License Server failsafe mode, by the Wrapper.
License file generator	The FLEXwrap program that allows the FLEXwrap license administrator at your site to generate FEATURE lines for wrapped applications. swsetup32.exe or swcrypter.exe This program should be protected from execution by your general user community. Otherwise, your users would be able to regenerate your license files, possibly compromising the integrity of your license management system.
License key	The 12- or 20-digit authentication code on a FEATURE line in a license file. Also called encryption code.
License server	The combination of one running lmgrd process and one running swrap vendor daemon process that allows wrapped applications to request and obtain licenses.
License server failsafe mode	A mode of operation that allows the Wrapper to run the wrapped application when the FLEXwrap License Server is down or cannot be contacted. This mode is enabled by specifying failsafe mode when the license is generated and making sure the license file is accessible to the Wrapper even when the License Server is down. See Setting Up Fail Safe Modes for more information.
License server machine	A computer system that is running both lmgrd and the swrap daemon. A copy of the license file will be on this machine, as will the FLEXwrap report log if turned on.
lmgrd	The license manager daemon process that starts the swrap vendor daemon and directs the Wrapper to the swrap daemon: lmgrd.exe
swrap	The FLEXwrap vendor daemon process that serves licenses to wrapped applications and writes a report log that FLEXnet Manager uses to generate reports: swrap.exe
Suite	A group of products sold by a software vendor where, according to the license agreement, each individual product cannot be used independently of the product suite.

Table 21-17 • FLEXwrap Terminology (cont.)

Term	Description
Wrapper	<p>The FLEXwrap binary which is run when an end user attempts to run a wrapped application:</p> <p>swrap_nt.exe (GUI apps)</p> <p>or</p> <p>swrap_ntc.exe (console apps)</p> <p>The Wrapper requests a license from the License Server and runs the wrapped application if a license is obtained or if license failsafe mode or License Server failsafe mode is enabled.</p>

About FLEXwrap License Parameters

This section lists the field name and explanation of each license parameter, and includes an example of a license file using each of these parameters. It also includes a list of all of the possible elements of a FLEXwrap FEATURE line.

- [Specifying FLEXwrap License Parameters](#)
- [Understanding FLEXwrap License File FEATURE Lines](#)

Specifying FLEXwrap License Parameters

To generate FLEXwrap licenses, you use the swsetup32 program and specify parameters using a FEATURE line. The following table lists the field name and explanation of each license parameter.

Table 21-18 • FLEXwrap License Parameters

Parameter	Description
Application Program	<p>This parameter must be identical (including case and excluding .exe) to the name of the binary to be wrapped. It must conform to FLEXwrap feature-naming rules, including:</p> <ul style="list-style-type: none">• The name must be <= 30 characters.• The name must start with either a letter, a number, or an underscore “_”. <p>This will appear after the word FEATURE on the FEATURE line.</p>

Table 21-18 • FLEXwrap License Parameters (cont.)

Parameter	Description
License Limit	<p>This parameter is the license limit—the number of licenses that the License Server will allow to be checked out simultaneously, not including overdraft.</p> <p>To strictly enforce the license limit for this feature, enter the number of licenses that you are licensed to run. If you do not wish to restrict licensing, but you want complete reports on all usage of the wrapped application, enter the largest number of instances of this binary that you think will be run simultaneously.</p> <p>If you are just testing FLEXwrap's functionality, you might want to enter 2 here. The license count will appear after the expiration date on the FEATURE line. For uncounted licenses, enter 0 or uncounted for the license limit.</p>
Overdraft	<p>This parameter is the number of licenses above the license limit that the License Server will allow to be checked out. Use this parameter if you want usage reports on usage and do not want to enforce license limits. Reports will show the license limit as the maximum number of licenses available, and any usage exceeding this limit will be overdraft usage.</p>
Program Suite	<p>Some vendors sell their products as a suite. FLEXwrap defines a suite as a group of independent programs which must, for licensing purposes, be treated as a unit.</p> <p>For example, if a product suite, Office, consists of programs Word, Paint, and Mail, and if you have only one license for the suite, then while one user was using Word, that same user could use products Paint or Mail, but a different user could not use any products—even Paint or Mail, which are not currently in use.</p> <p>Typically, to create the licenses for a product suite, you would create licenses with the Group by (duplicate license request) parameter set so that multiple requests from the same user (or the same user on the same host) count as one license request.</p> <p>When you create a suite license with FLEXwrap, the Wrapper checks out both the individual program license (Word in the example above) and a suite license (Office in the example). Since a second license request for Office does not require an additional license, when the same user runs program Paint, he is able to check out the Office suite license, and is also allowed to check out the Paint license. If another user attempts to check out a license for Paint, she would be unable to get the Office suite license and her request would be denied (or queued).</p> <p>An application is made part of a suite by using the optional PREREQ attribute of a FLEXwrap license FEATURE line. PREREQ is set to a FEATURE that must also be checked out before the wrapped application will be executed.</p> <p>To create a suite you must add a PREREQ line to all FEATURES that will be part of the suite. You must then create the suite's FEATURE in the license file. A full suite FEATURE looks like:</p> <pre>FEATURE suite name swrap 1.0 expiration count 0 \ DUP_GROUP=[UHDS]</pre> <p>This is identical to an application's FEATURE line except that only the DUP_GROUP optional attribute is pertinent.</p>

Table 21-18 • FLEXwrap License Parameters (cont.)



Parameter	Description
Expiration Date	The date the license for this FEATURE will expire. Licenses can no longer be obtained after this date.
Node-lock Hostid	This is the hostid of the only machine(s) on which the wrapped application can run. This field can contain multiple hostids separated by spaces and surrounded by double quotes.
Application File to Run	Unless you specify otherwise, when the Wrapper obtains a license, it will run <code>binary.orig</code> (or <code>binary.fwr</code>) in the directory from which the Wrapper ran. If you want it to run a different binary, specify the full path to the binary name here. This will become part of the FEATURE line as <code>w_binary=full_path_to_binary</code> .
Queueing Behavior	<ul style="list-style-type: none"> • Queue/Always—Queue for a license if one cannot be obtained. When a license becomes available, the binary will run. This is for use when failsafe is not on. This is recommended for batch jobs. (<code>w_queue=1</code>) • Query/Ask—If a license cannot be obtained, shows who is using the licenses and asks the user if they want to queue for a license. If the user answers Y, the Wrapper will wait until a license is available. When one becomes available, the binary will run. (<code>w_queue=2</code>) • Exit/Never—No queueing.
Duplicate Grouping	<p>FLEXwrap allows you to specify that simultaneous uses of the wrapped application by the same user (or host, or user/host combination, etc.) will not require a license.</p> <p>The default is that each process requires its own license. To accept this default, in <code>swsetup32</code>, choose None.</p> <p>You can group on:</p> <ul style="list-style-type: none"> • User—All requests from the same user use only one license. • Host—All requests from the same host use only one license. • User + Host—All requests from the same user on the same host use one license. • Site—All requests use one license.
Regular (License) Failsafe Mode	<p>If you wish to strictly enforce your license limits, check in <code>swsetup32</code>. If a license cannot be obtained, the binary will not run. This will not appear on the FEATURE line.</p> <p>If you want the binary to run even if it cannot obtain a license, uncheck in <code>swsetup32</code>. This will appear as <code>VENDOR_STRING=failsafe</code> on the FEATURE line.</p>  <p>Note • For License Server failsafe mode, where the application will never fail to run because it cannot connect to the FLEXwrap License Server, the Wrapper must be able to see the license file when the License Server is down. See Setting Up License Server Failsafe Mode for instructions.</p>

Table 21-18 • FLEXwrap License Parameters (cont.)

Parameter	Description
Generate 20-Digit License Keys	<p>The default length of the license key on a FEATURE line generated by FLEXwrap v8.2 is 12 digits.</p> <ul style="list-style-type: none"> • If you run swsetup32 on Windows or run swcrypter from the command line, you can generate a 20-digit license key with embedded start date for backward compatibility. • A FLEXwrap v8.2 license file can contain a mixture of 12- and 20-digit license keys. • To generate 20-digit keys, set the license format to FLEXlm v5 in swsetup32 or swcrypter.
Shutdown Signal	<p>Some applications require a special shutdown signal in order to perform application-specific cleanup. FLEXwrap will normally send a SIGHUP to the application, but if the application requires a different signal, you can specify it in this parameter. For example, vi will leave the terminal in “raw” mode unless it receives a SIGHUP.</p>
Connection Timeout	<p>If the License Server becomes unavailable (due to a License Server node failure, a License Server process failure, or a network communication problem), the application will lose its license.</p> <p>The Wrapper will attempt to regain its license until the timeout (or exit) period elapses. By default, there is no timeout (or exit) period, so the application will continue to run without a license and will continue to attempt to regain its license indefinitely.</p> <p>If you specify a timeout period, the wrapped application will continue to run for the specified amount of time while the Wrapper attempts to regain its license. The Wrapper will give periodic warnings to the user. For example, if you set the timeout to ten minutes, the user will get ten messages like this:</p> <pre>Lost license for feature. Reconnection attempt n of 10.</pre> <p>If a license can be obtained before the timeout period ends, the user will see Reacquired feature license. If the timeout period is reached and a license has not been obtained, the application will exit with the message Lost license. Exiting.</p>

Table 21-18 • FLEXwrap License Parameters (cont.)

Parameter	Description
Runtime Name	<p>Some application programs require that their argv[0] is different from the last part of the application file name. To accommodate these rare cases, FLEXwrap allows you to specify what value to pass as argv[0] to the application.</p> <p>For example, the vi program will only operate in what would be considered a “normal” manner if its argv[0] is vi. If you were to rename the binary for vi to /usr/ucb/vi_real, for example, then you would want to specify vi as the argv[0] parameter.</p> <p>Another reason to use the argv[0] parameter would be for programs that do an internal restart by calling the exec() function. When the function which causes the product to call exec() is executed, the exec() call causes a second copy of the FLEXwrap Wrapper to run, thereby checking out a second license.</p> <p>By setting argv[0] to program.orig (and using the standard renaming convention of renaming program to program.orig), then the exec() call executes the real program rather than the Wrapper, and the second license is not checked out.</p> <div><p>Note • If the user does not have the program in his path, then the argv[0] parameter will need to be the full path name of the wrapped program. This will not work in the case where the Wrapper is used to wrap multiple architecture binaries (since they will be in different directories and only one directory can be specified to argv[0] in this manner). In this case, the user will need to place the binary directory into his shell path and set argv[0] to the program name, without the path. There are very, very few cases in Windows when this setting is needed.</p></div>



Note • See [Understanding FLEXwrap License File FEATURE Lines](#) for additional explanation of these parameters.

A FEATURE line with all possible attributes enabled would look like this:

```
FEATURE feature_name swrap 1.0 expiration_date count 0 \  
  w_binary=full_path_to_binary \  
  w_argv=argv \  
  PREREQ=suite_name \  
  W_LIC_LOSS=#minutes \  
  VENDOR_STRING=failsafe \  
  HOSTID=node_lock_hostid \  
  DUP_GROUP=[U, H, UH, S] \  
  w_queue=[not set, 1, 2] \  
  w_term_signal=signal#
```

Understanding FLEXwrap License File FEATURE Lines

The simplest FEATURE line in a FLEXwrap license file will look something like this:

```
FEATURE testvi2 swrap 1.000 1-jan-0000 2 BB8C401190F6
```

The most complex FEATURE line, in which all the options have been exercised, will look something like this:

```
FEATURE featurea swrap 1.000 15-dec-2001 2 6B7C60F1803D \
  VENDOR_STRING=failsafe HOSTID=1234 \
  DUP_GROUP=U w_binary=/t/dodi/real w_argv=real w_queue=1 \
  w_term_signal=2 W_LIC_LOSS=5 PREREQ=suite
```

Here are all the possible elements of a FLEXwrap FEATURE line. Keywords in upper case letters are part of the license encryption, so they cannot be manually edited without corrupting the FEATURE line. Those in lower case are not part of the encryption, so they can be changed manually without regenerating the license.

Table 21-19 • FLEXwrap FEATURE Line Elements

Element	Possible Values	Explanation
FEATURE	FEATURE	
<i>feature_name</i>	Same as wrapped application's original binary name (including case).	
<i>daemon_name</i>	swrap	
<i>version</i>	1.000	
<i>expiration_date</i>	Any valid date in the format dd-mm-yy[yy]. <ul style="list-style-type: none"> Any year shown as two digits is assumed to be 19xx. For example: 31-dec-99 is December 31, 1999. Any year after 1999 must be shown with four digits. For example: 1-jan-2000. Unexpiring can be shown as: 1-jan-0 or 1-jan-00 or 1-jan-0000, permanent. 	
<i>count</i>	Any positive integer. For uncoun­ted licenses, the count is set to 0 or uncoun­ted.	The number of licenses that can be checked out simultaneously without going into overdraft. This number will appear as the license limit on reports.

Table 21-19 • FLEXwrap FEATURE Line Elements (cont.)

Element	Possible Values	Explanation
license_key	12- or 20-digit hexadecimal number.	<p>FLEXwrap license authorization key generated by the FLEXwrap license file generator.</p> <p>12-digit license keys are generated by default. A FLEXwrap v8.2 license file can contain a mixture of 12- and 20-digit license keys.</p> <p>For backward compatibility, 20-digit license keys containing an embedded start date can be generated using <code>swsetup32</code> on Windows or by using an option to <code>swcrypter</code>:</p> <pre>swcrypter license_file -verfmt 5</pre>
VENDOR_STRING=	failsafe	<p>In regular (license) failsafe mode, the wrapped application specified on the FEATURE line can run without obtaining a license if the Wrapper can connect to the License Server.</p> <p>To set up License Server failsafe mode in which the wrapped application can run even if the License Server is not running, see Setting Up License Server Failsafe Mode</p>
HOSTID=	node_lock_hostid	<p>The only machine(s) where the wrapped application can run.</p> <p>Multiple hostids are separated by spaces and have double quotes around the string.</p>
DUP_GROUP=	<ul style="list-style-type: none"> • U—User • H—Host • D—Display • UH—User and Host • UD—User and D • UHD—User, Host and D • HD—Host and D • S—Site 	<p>Additional simultaneous instances by the same (U)ser, (H)ost, (U)ser & (H)ost, (S)ite, etc. will not be counted against the license limit.</p>

Table 21-19 • FLEXwrap FEATURE Line Elements (cont.)

Element	Possible Values	Explanation
w_binary=	<i>full_path_to_binary</i>	Binary to run when license has been obtained.
w_argv=	<i>argv</i>	When wrapped, the application binary's name is changed to <i>binary.orig</i> . Some applications do not work properly unless their name is <i>binary</i> . In those cases, specify <i>binary</i> as the <i>argv</i> .
w_queue=	1 2	If license cannot be obtained, and failsafe is not set: 1 = Queue automatically 2 = Ask if user wishes to queue
w_term_signal	<i>signal#</i>	Signal sent to wrapped application when the Wrapper program is killed. (default=1)
W_LIC_LOSS=	<i>#minutes</i>	This is the timeout value. If a timeout is specified, and if the Wrapper loses its license, the Wrapper will warn the user and exit after this number of minutes. The minimum connection timeout is five minutes.
PREREQ=	<i>suite_name</i>	Feature name of suite license that must also be obtained in order for this license to be obtained.

Dialog Boxes and Wizards

The FLEXwrap Server Configuration Tool uses the following dialog boxes and Wizards:

- [FLEXwrap Server Configuration Dialog Box](#)
- [FLEXwrap Application Information Dialog Box](#)
- [Wrapping Wizard](#)
- [Encryption Wizard](#)
- [Wrapping and Encryption Wizard](#)

- [FLEXwrap Command Dialog Box](#)
- [Invalid Certificate Dialog Box](#)

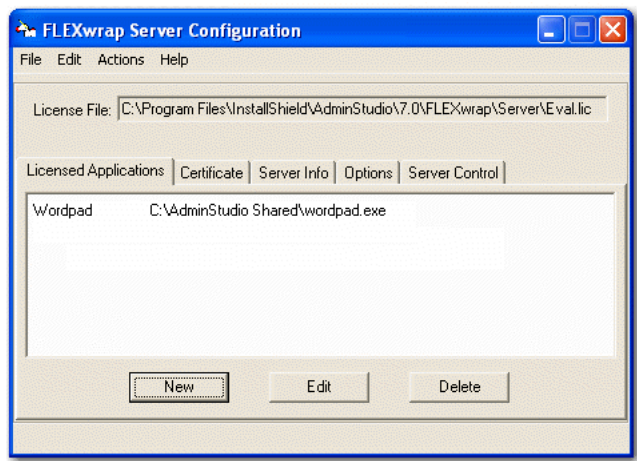
FLEXwrap Server Configuration Dialog Box

The **FLEXwrap Server Configuration** dialog box includes the following tabs:

- [Licensed Applications Tab](#)
- [Certificate Tab](#)
- [Server Info Tab](#)
- [Options Tab](#)
- [Server Control Tab](#)

Licensed Applications Tab

The **Licensed Applications** tab lists all of the licensed applications in the currently selected License File.



The following options are included:

Table 21-20 • Server Configuration Tool Dialog Box—Licensed Applications Tab

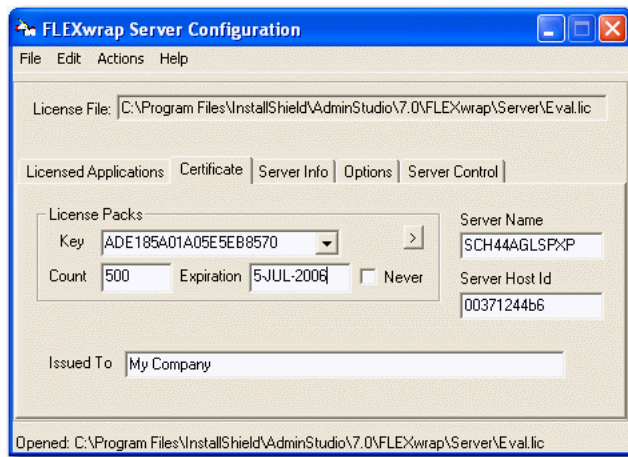
Option	Description
License File	The location of the selected License File. To open a License File, click Open on the File menu.
Licensed Applications	A list of all of the licensed applications in the currently selected License File.
New	Click to open the FLEXwrap Application Information Dialog Box where you select an application executable file to FLEX-enable.

Table 21-20 • Server Configuration Tool Dialog Box—Licensed Applications Tab

Option	Description
Edit	Select a listed application and click Edit to edit that application's settings on the FLEXwrap Application Information Dialog Box .
Delete	Click to delete a licensed application from the list.

Certificate Tab

On the **Certificate** tab, you enter the production FLEXwrap license information you received from AdminStudio Support for an application.



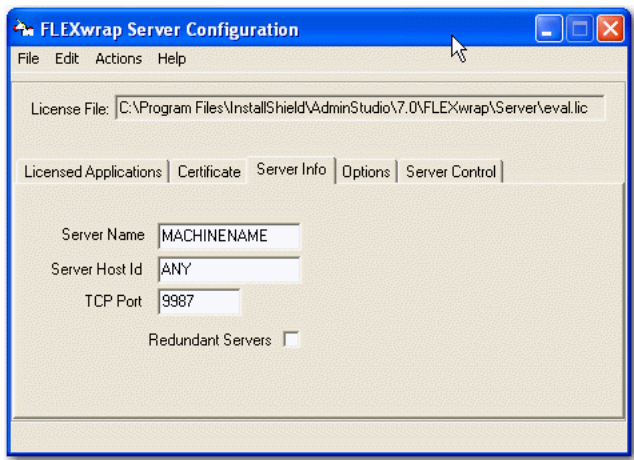
The following options are included:

Table 21-21 • Server Configuration Tool Dialog Box—Certificate Tab

Option	Description
Key	License key provided by AdminStudio Support.
Count	The number of concurrent licenses available for this FLEX-enabled application.
Expiration	The date these licenses will expire.
Never	Select if you do not want these license to expire.
Server Name	Name of the FLEXwrap License Server machine.
Server Hostid	The host name of the FLEXwrap License Server listed in the production license file.
Issued To	The name of the company that purchased the license.

Server Info Tab

This tab lists the identification information for the FLEXwrap License Server.



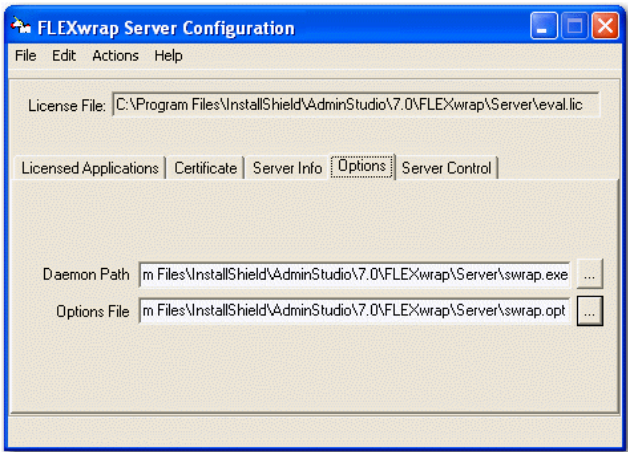
The following options are included:

Table 21-22 • Server Configuration Tool Dialog Box—Server Info Tab

Option	Description
Server Name	Name of the FLEXwrap License Server
Server Host Id	Hostid of the FLEXwrap License Server.
TCP Port	TCP Port of the FLEXwrap License Server.
Redundant Servers	Select this option if there are any redundant (back-up) servers associated with the specified server.

Options Tab

On the Options tab, you can set the path for the Daemon and the Options File.



The following options are included:

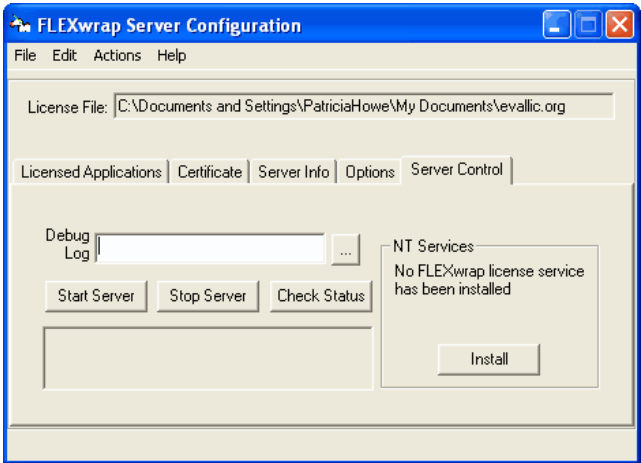
Table 21-23 • Server Configuration Tool Dialog Box—Options Tab

Option	Description
Daemon Path	If your swrap daemon is located in the same folder as the FLEXwrap Server Configuration tool (lmgrd.exe), you can leave the Daemon Path field blank. Otherwise, enter the full path to the swrap daemon.
Options File	<p>If you want to record license usage in a report log, create a FLEXwrap options file. You could then generate usage reports using FLEXnet Manager.</p> <p>Create a plain text file called swrap.opt and save it in the same directory with your license file. In the Options File field, enter the full path to this options file. See Creating an Options File for more information.</p>

Server Control Tab

On the Server Control tab, you can:

- Configure the FLEXwrap License Server to run as a service.
- Manually start the FLEXwrap License Server.
- Stop the FLEXwrap License Server.
- Check the status of the FLEXwrap License Server.



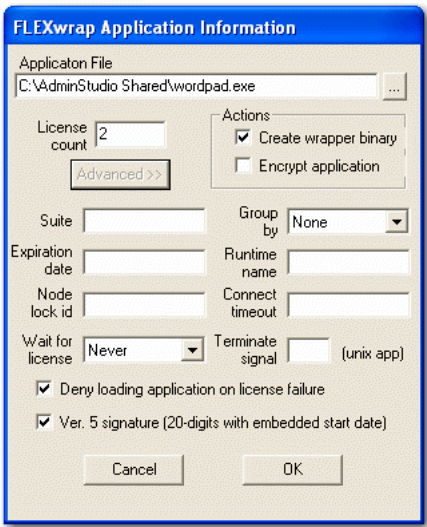
The following options are included:

Table 21-24 • Server Configuration Tool Dialog Box—Server Control Tab

Option	Description
Debug Log	To manually start the FLEXwrap License Server, you first need to enter the path to a debug log or select an existing debug log. If you do not specify a debug log when you start the License Server, at least two console windows will appear—one for the lmgrd output, one for the swrap output, and one for each of any other vendor daemons started by this lmgrd .
Start Server	To start the FLEXwrap License Server, click the Start Server button. All licenses (*.lic) in the directory of the opened license are loaded.
Stop Server	To stop the FLEXwrap License Server, click the Stop Server button.
Check Status	To check the status of the FLEXwrap License Server, click the Check Status button A Status Report opens.
Install	To configure the FLEXwrap License Server to run as a service, click the Install button in the NT Services area. The selected license file is now configured to run as a service.

FLEXwrap Application Information Dialog Box

On the FLEXwrap Application Information dialog box, you select the application binary that you are going to wrap, enter the license limit for the application, and set other advanced options.



The following options are included:

Table 21-25 • FLEXwrap Application Information Dialog Box Options—Advanced View

Option	Description
Application File	Select the application binary that you are going to wrap. Unless you specify otherwise, when the Wrapper obtains a license, it will run <code>binary.orig</code> (or <code>binary.fwr</code>) in the directory from which the Wrapper ran. If you want it to run a different binary, specify the full path to the binary name here. This will become part of the FEATURE line as <code>w_binary=full_path_to_binary</code> .
Create wrapper binary	Make sure this option is selected.
Encrypt application	In some cases, the application binary can be encrypted so that it can be run only by the Wrapper. This would prevent users from running the wrapped application binary (<code>binary.fwr</code>) directly. If you want to encrypt the application binary, select the this option.

Table 21-25 • FLEXwrap Application Information Dialog Box Options—Advanced View (cont.)

Option	Description
License Count	<p>Enter the license limit for the application—the number of licenses that the License Server will allow to be checked out simultaneously, not including overdraft.</p> <ul style="list-style-type: none"> • To strictly enforce the license limit for this feature, enter the number of licenses that you are licensed to run. • If you do not wish to restrict licensing, but you want complete reports on all usage of the wrapped application, enter the largest number of instances of this binary that you think will be run simultaneously. • For uncounted licenses, enter 0 or uncounted for the license limit. • If you are just testing FLEXwrap's functionality, you might want to enter 2 here. • The license count will appear after the expiration date on the FEATURE line. • The license limit will appear in reports as the maximum number of licenses available. • The total of the counts of licenses on all the FEATURE lines you generate may not exceed the count on the FLEXwrap INCREMENT line in your license file.
Advanced	Click to display advanced options.
Suite	<p>If you want to treat the selected executable files as a suite, enter a suite name in this field.</p> <p>A suite is a group of independent programs which must, for licensing purposes, be treated as a unit. Only one license is required for one person to use all of the applications in the suite.</p> <p>An application is made part of a suite by using the optional PREREQ attribute of a FLEXwrap license FEATURE line. PREREQ is set to a FEATURE that must also be checked out before the wrapped application will be executed.</p> <p>To create a suite you must add a PREREQ line to all FEATUREs that will be part of the suite. You must then create the suite's FEATURE in the license file. A full suite FEATURE looks like:</p> <pre>FEATURE suite name swrap 1.0 expiration count 0 \ DUP_GROUP=[UHDS]</pre> <p>This is identical to an application's FEATURE line except that only the DUP_GROUP optional attribute is pertinent.</p>
Expiration date	The date the license for this FEATURE will expire. Licenses can no longer be obtained after this date.
Node lock id	This is the hostid of the only machine(s) on which the wrapped application can run. This field can contain multiple hostids separated by spaces and surrounded by double quotes.

Table 21-25 • FLEXwrap Application Information Dialog Box Options—Advanced View (cont.)

Option	Description
Wait for license	<p>Select one of the following options to define the queuing behavior of this wrapped application:</p> <ul style="list-style-type: none"> • Always—Queue for a license if one cannot be obtained. When a license becomes available, the binary will run. This is for use when failsafe is not on. This is recommended for batch jobs. (w_queue=1) • Ask—If a license cannot be obtained, shows who is using the licenses and asks the user if they want to queue for a license. If the user answers Y, the Wrapper will wait until a license is available. When one becomes available, the binary will run. (w_queue=2) • Never—No queueing.
Group by	<p>FLEXwrap allows you to specify that simultaneous uses of the wrapped application by the same user (or host, or user/host combination, etc.) will not require a license.</p> <p>The default is that each process requires its own license. To accept this default, in swsetup32, choose None.</p> <p>You can group on:</p> <ul style="list-style-type: none"> • User—All requests from the same user use only one license. • Host—All requests from the same host use only one license. • User + Host—All requests from the same user on the same host use one license. • Site—All requests use one license.
Runtime name	<p>Some application programs require that their argv[0] is different from the last part of the application file name. To accommodate these rare cases, FLEXwrap allows you to specify what value to pass as argv[0] to the application by entering a name in this box. For more information on specifying a Runtime name, see Runtime Name.</p>

Table 21-25 • FLEXwrap Application Information Dialog Box Options—Advanced View (cont.)

Option	Description
Connect timeout	<p>If the License Server becomes unavailable (due to a License Server node failure, a License Server process failure, or a network communication problem), the application will lose its license.</p> <p>The Wrapper will attempt to regain its license until the timeout (or exit) period elapses. By default, there is no timeout (or exit) period, so the application will continue to run without a license and will continue to attempt to regain its license indefinitely.</p> <p>If you specify a Connect timeout period, the wrapped application will continue to run for the specified amount of time while the Wrapper attempts to regain its license. The Wrapper will give periodic warnings to the user. For example, if you set the timeout to ten minutes, the user will get ten messages like this:</p> <p>Lost license for feature. Reconnection attempt n of 10.</p> <p>If a license can be obtained before the timeout period ends, the user will see Reacquired feature license. If the timeout period is reached and a license has not been obtained, the application will exit with the message Lost license. Exiting.</p>
Terminate signal	<p>Some applications require a special shutdown signal in order to perform application-specific cleanup. FLEXwrap will normally send a SIGHUP to the application, but if the application requires a different signal, you can specify it in this parameter. For example, vi will leave the terminal in “raw” mode unless it receives a SIGHUP</p>
Deny loading application on license failure	<p>By default, the Deny loading application on license failure option is selected.</p> <ul style="list-style-type: none"> • Run when no license is available—If you want the wrapped application to be able to run whether or not it can get a license from a running FLEXwrap License Server (License Failsafe Mode), clear the Deny loading application on license failure option. See Setting Up Regular (License) Failsafe Mode. • Run when cannot connect to License Server—If you want the wrapped application to be able to run whether or not the FLEXwrap License Server is running (License Server Failsafe Mode), clear the Deny loading application on license failure option and follow the instructions in Setting Up License Server Failsafe Mode. • Cannot run without a license—If you wish to require the Wrapper to obtain a license in order to run the wrapped application, select the Deny loading application on license failure option

Table 21-25 • FLEXwrap Application Information Dialog Box Options—Advanced View (cont.)

Option	Description
Ver. 5 signature (20-digits with embedded start date)	<p>Select this option if you want to generate 20-digit license keys with an embedded start date.</p> <ul style="list-style-type: none"> • The default length of the license key on a FEATURE line generated by FLEXwrap v8.2 is 12 digits. • A FLEXwrap v8.2 license file can contain a mixture of 12- and 20-digit license keys. • If you run swsetup32 on Windows or run swcrypter from the command line, you can generate a 20-digit license key with embedded start date for backward compatibility.

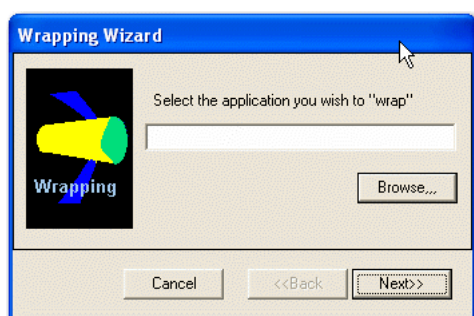
Wrapping Wizard

The Wrapping Wizard includes the following panels:

- [Select File to Wrap Panel](#)
- [Identify the License Server and License File Panel](#)
- [Identify Location of Original Binary Panel](#)
- [Ready to Wrap Panel](#)

Select File to Wrap Panel

Select the application executable (.exe) that you wish to wrap and click **Next**.



Identify the License Server and License File Panel

On this panel, you are prompted to identify the path to the License Server. You can choose to enter a host and port number or a path to the location of the license file.



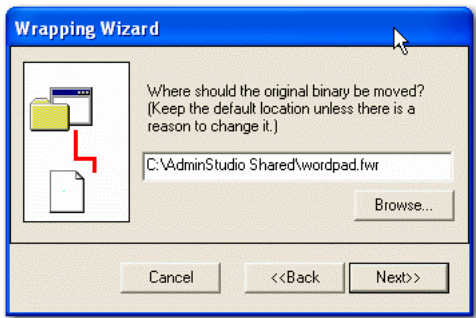
The following options are included:

Table 21-26 • Wrapping Wizard—License File Location Panel

Option	Description
Host and port number reference	If you select this option, enter the host and port number of the license server.
Path to the license file	If you select this option, enter the appropriate path to the license server. This information is written into swrapper .1f. You would normally want to select port@host .

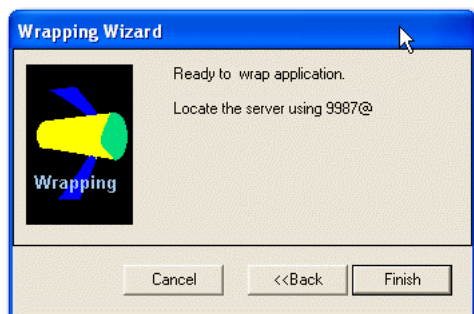
Identify Location of Original Binary Panel

On this panel, you are prompted to identify the location where you want FLEXwrap to save a copy of the original binary prior to wrapping.



Ready to Wrap Panel

To begin wrapping the selected package, click **Finish**.



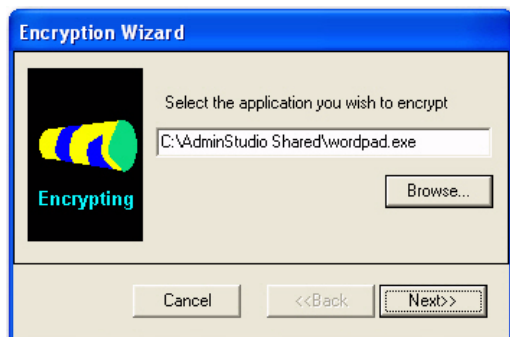
Encryption Wizard

The Encryption Wizard includes the following panels:

- [Select Application Panel](#)
- [Location for Original Binary Panel](#)
- [Ready to Encrypt Panel](#)

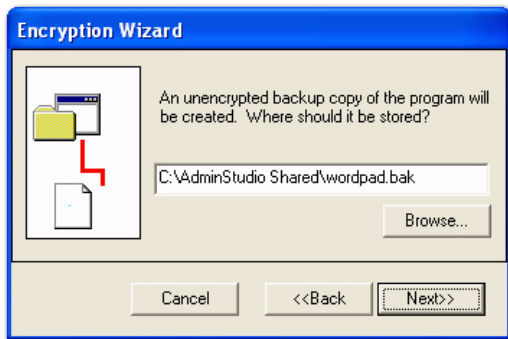
Select Application Panel

Select the application executable (.exe) that you wish to encrypt and click **Next**.



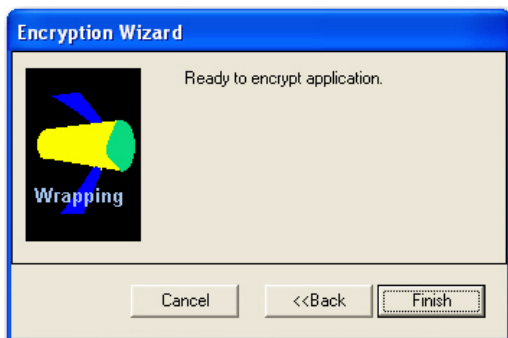
Location for Original Binary Panel

On this panel, you are prompted to identify the location where you want FLEXwrap to save a copy of the original binary prior to encryption.



Ready to Encrypt Panel

To begin encryption of the selected package, click **Finish**.



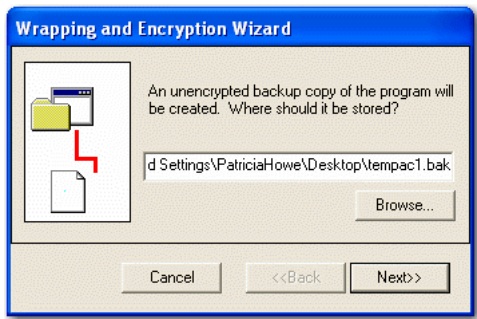
Wrapping and Encryption Wizard

The Wrapping and Encryption Wizard includes the following panels:

- [Backup Copy Panel](#)
- [License File Location Panel](#)
- [Original Binary Location Panel](#)
- [Ready to Encrypt and Wrap Panel](#)

Backup Copy Panel

On this panel you specify where you would like a backup copy of the application binary to be saved. The default is `binary.bak` in the application directory



License File Location Panel

On the License File Location Panel, you are prompted to identify the path to the License Server. You can choose to enter a host and port number or a path to the location of the license file.



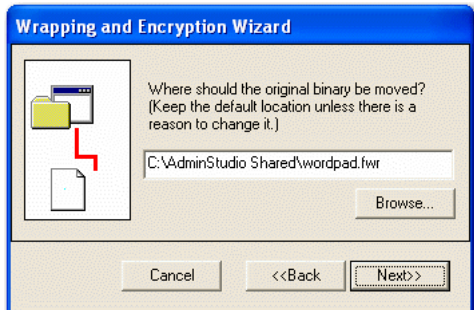
The following options are included:

Table 21-27 • Wrapping and Encryption Wizard—License File Location Panel

Option	Description
Host and port number reference	If you select this option, enter the host and port number of the license server.
Path to the license file	If you select this option, enter the appropriate path to the license server. This information is written into <code>swrapper.1f</code> . You would normally want to select port@host .

Original Binary Location Panel

On this panel you are prompted to identify the location where you want FLEXwrap will save a copy of the original binary prior to encryption.



Ready to Encrypt and Wrap Panel

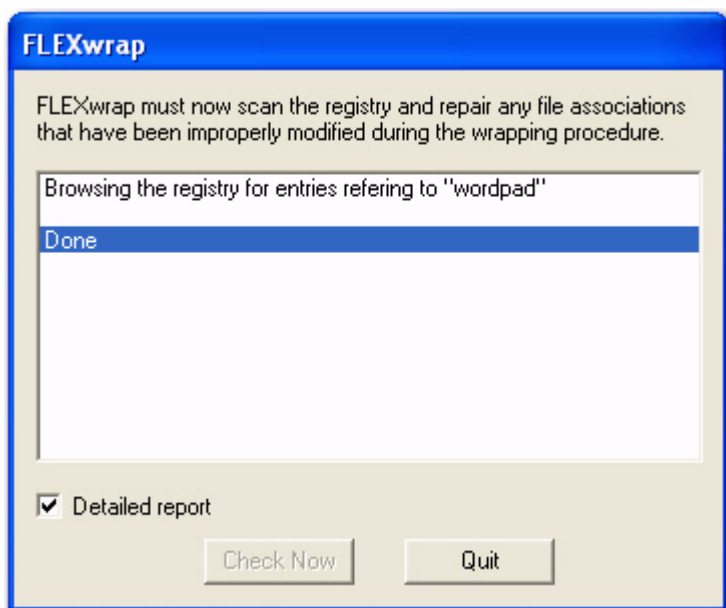
To begin encryption and wrapping of the selected package, click **Finish**.



FLEXwrap Command Dialog Box

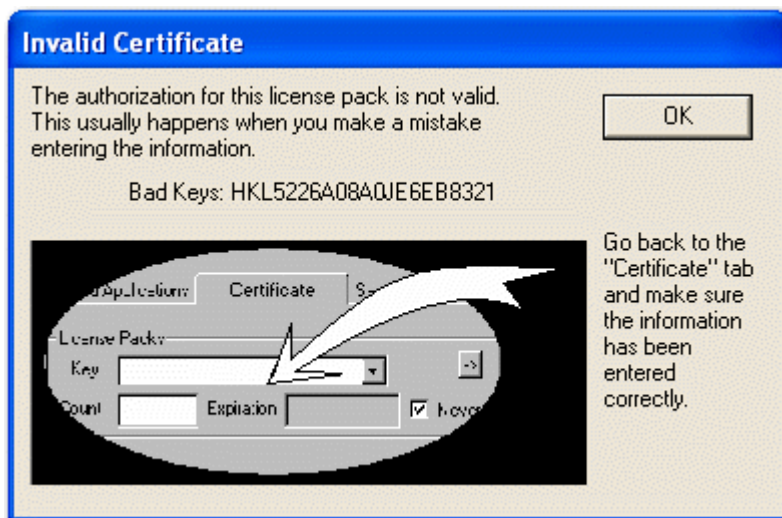
When the [Wrapping Wizard](#) is finished, this dialog box opens, informing you that FLEXwrap must now scan the Registry and repair any file associations that were improperly modified during the wrapping procedure.

Click **Check No** to scan/repair the registry. To get a detailed report of the Registry scan, select the **Detailed report** option.



Invalid Certificate Dialog Box

This dialog box opens when an invalid license key was entered in the License Packs **Key** field on the [Certificate Tab](#) of the [FLEXwrap Server Configuration Dialog Box](#). You should check to see whether the **Key** was entered correctly.



You purchase production license file credentials from AdminStudio Support, and then create the license file by entering the license file credentials on the [Certificate Tab](#).

Part 6

Using AdminStudio Enterprise Server Tools

This part of the User Guide includes the following chapters:

- [Generating and Viewing Reports in Report Center](#)
- [Automating Tasks Using Job Manager](#)
- [AdminStudio.Public API Reference](#)

Generating and Viewing Reports in Report Center



Edition • Report Center is included with AdminStudio Enterprise Edition

Report Center provides a centralized view of all of the information regarding packages in your Application Catalog. You can use Report Center to generate reports on packages stored in the Application Catalog. You can also use SQL queries to generate custom reports on data stored in the Application Catalog.

- **Package Report**—Includes consolidated package information including conflicts, validation results, and extended attributes. See [Viewing Package Reports](#).
- **Custom SQL Query Report**—A custom report defined by entering an SQL query in the Report Wizard. See [Generating a Custom SQL Query Report for AdminStudio](#).

Generating and Viewing AdminStudio Reports

You can use Report Center to obtain a centralized view of all of the information regarding packages in your AdminStudio Application Catalog. Because Report Center is a Web application, it can be easily accessed by a geographically dispersed workforce without requiring any software installation or data transfer. Report Center makes it easy to get the application data you need to diagnose and repair software problems and to manage applications across your organization.

A catalog-level search tool enables you to generate detailed, customizable reports on packages with particular characteristics. These reports are accessible anywhere via a Web interface and can be exported to PDF or Excel format for sharing and archiving.

Information on generating and viewing AdminStudio reports in Report Center is presented in the following sections:

Table 22-1 • Information About Generating AdminStudio Reports

Section	Description
About the Package Report Tab of the Report Center	Explains the layout of the Select Package page on the Package Report tab, and the Package and Group summary information that can be viewed on this page.
Generating a Package Report	Explains how to generate a Package Report on a selected package. Also explains how to filter the package tree by specified criteria in order to find a specific package in the Application Catalog.
Information Included in Package Reports	Lists the contents of all of the sections of a Package Report.
Archiving a Package Report	Explains how to create a PDF-snapshot of a Package Report at a particular date and time and how to open an archived report.
Exporting a Package Report	Explains how to export a Package Report to Excel or PDF format.
Generating a Custom SQL Query Report for AdminStudio	Explains how to enter an SQL query to specify the data to be displayed in a custom report.

About the Package Report Tab of the Report Center

You generate most AdminStudio reports on the **Select Package** page on the **Package Report** tab in Report Center.

Select Package

Browse and select a package from the Packages tree. Click View Report to generate a detailed report for the selected package.

Groups

- Adobe Acrobat 4.05
- Adobe Acrobat 6.0 Professional
- Adobe Photoshop
- Illustrator 10**
- Microsoft AntiSpyware
- Windows2000

Show All Packages

Illustrator 10 View Report

Product Version	6.22
Manufacturer	Adobe
Import Date	10/16/2006 8:27:40 PM
In Software Repository	No
Unresolved Conflicts	0
Associated Application Request	

Search Packages Search Clear All

Enter values for one or more search criteria and click the Search button.

Package Attributes

Product Name:	Manufacturer:	Product Version:	Package Code:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Product Code:	Upgrade Code:	Setup File Name:	Comments:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Import Date:	Extended Attributes:		
<input type="text"/>	<input type="text"/>		

Package Content

Application Request Attributes

Name:	Upload Date:	Due Date:	Risk Date:
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Due Period:	End Date:		
<input type="text"/>	<input type="text"/>		

Figure 22-1: Select Package Page of Report Center

The **Select Package** page is organized in the following sections:

Table 22-2 • Areas on the Select Package Page

Area	Description
Package Tree	On the left side of the interface, all of the groups and packages in the AdminStudio Enterprise Server Application Catalog are displayed in a tree control. If a search has been applied using the fields in the Search Packages area, only those packages meeting that criteria are listed, and the Show All packages button is enabled.
Package or Group Summary Area	In the top half of the right side of the interface, either a summary of group statistics (when a group is selected) or a summary of package statistics (when a package or OS Snapshot is selected) is displayed. See Group Summary Information and Package Summary Information . When a Windows Installer .msi package is selected, you can click View Report to view that package's Package Report.
Search Packages Area	In bottom half of the right side of the interface, the Search Packages area includes fields allowing you to search for packages by attributes, content, or associated Application Requests.

Group Summary Information

When a group in the package tree is selected, summary information about that group is listed.


 Groups	
Total packages in group	9
Windows Installer packages	5
OS Snapshot packages	1
Packages with conflicts	2

Figure 22-2: Group Summary on the Select Package Page

The following information about the selected group is listed:

Table 22-3 • Group Summary Information

Item	Description
Total Packages in Group	Total number of packages that have been imported into the selected group. This number is also broken down between number of Windows Installer packages and the number of OS Snapshot packages.

Table 22-3 • Group Summary Information

Item	Description
Packages With Conflicts	Number of packages that have unresolved conflicts with other packages or OS Snapshots in the Application Catalog.

Package Summary Information

When a package in the package tree is selected, summary information about that package is listed.

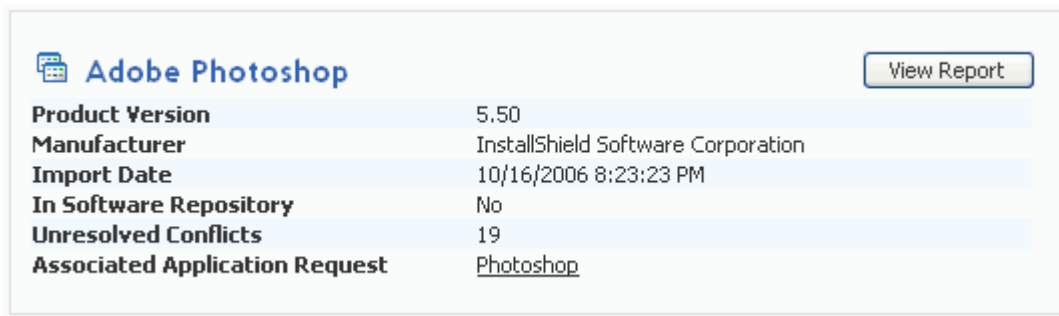


Figure 22-3: Package Summary on the Select Package Page

The following information about the selected package is listed:

Table 22-4 • Package Summary Information

Item	Description
Product Version	Version of package that is recorded in the package's Windows Installer file.
Manufacturer	Company that authored the package.
Import Date	The date and time the package was imported into the Application Catalog.
In Software Repository	Indicates whether or not this package and its associated files are managed by the Software Repository.
Unresolved Conflicts	The number of detected conflicts, generated during conflict analysis of this package, which have not yet been resolved—either automatically or manually.
Associated Application Request	If this package is associated with an Application Request, the Application Request name is listed here. Click this link to open the Application Request's Application Report.

Generating a Package Report

You generate a Package Report by selecting a package in the package tree and then clicking **View Report**. The package tree can list either all of the packages in the Application Catalog or you can filter the list of packages by specified criteria by using the fields in the **Search Packages** area.

- [Searching for a Package on the Select Package Page](#)
- [Viewing Package Reports](#)

Viewing Package Reports

You can view a Package Report by opening the **Select Package** page on the Report Center **Package Reports** tab, selecting a package from the package tree, and then clicking **View Report**.



Task: *To view a Package report:*

1. Open the **Package Report** tab of the **Report Center**. The **Select a Package** page opens.
2. Select a package in the package tree. The **Package Summary** information is listed.



Note • To filter the package tree by specific criteria, perform a search as described in [Searching for a Package on the Select Package Page](#).

3. Click **View Report**. The Package Report for the selected package opens.



Note • You can archive this report by clicking the **Archive Report** link in the lower left of the Package Report. See [Archiving a Package Report](#). You can also export it in PDF or Excel format. See [Exporting a Package Report](#).

Searching for a Package on the Select Package Page

On the **Select Package** page, you can filter the list of packages displayed in the package tree to display only those packages that meet specific search criteria, which are grouped into three categories:

- **Package Attributes**—Search by properties assigned to the Windows Installer package. See [Package Attributes](#).
- **Package Content**—Search by files, registry entries, .ini files, or shortcuts contained in the Windows Installer package. See [Package Content](#).
- **Application Request Attributes**—Search by information related to a package's associated Application Request. See [Application Request Attributes](#).

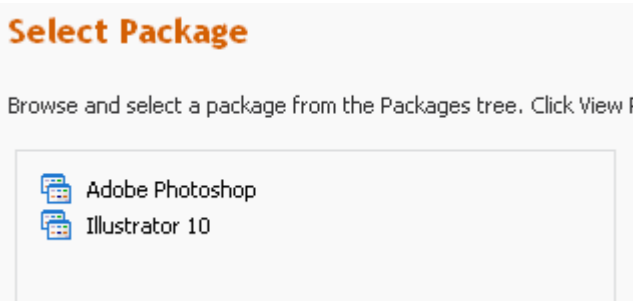
To filter the list of packages displayed in the package tree to display only those packages that meet specific search criteria, perform the following steps.



Task: *To search for a package on the Select Package page:*

1. In the **Search Packages** area of the **Select Package** page, expand the criteria category that you want to use by clicking the arrow. When all three categories are expanded, the following fields are available:

2. Enter values in the criteria fields that you want to search on. You can search for packages in the Application Catalog based on metadata in three categories:
 - **Package Attributes**—Search by properties assigned to the Windows Installer package. See [Package Attributes](#).
 - **Package Content**—Search by files, registry entries, .ini files, or shortcuts contained in the Windows Installer package. See [Package Content](#).
 - **Application Request Attributes**—Search by information related to a package's associated Application Request. See [Application Request Attributes](#).
3. After you have entered the search criteria, click **Search**. The packages that meet the criteria are now listed in the package tree in alphabetical order and are no longer grouped.




Package Attributes

You can search for packages in a catalog based on one or more of any of the following Package attribute metadata:

Table 22-5 • Package Attribute Search Fields

Metadata	Description
Product Name	Enter a package name to search by.
Manufacturer	Enter the name of the software vendor who authored the package.
Product Version	Enter the value that was entered in the Windows Installer ProductVersion property for this package.
Package Code	Enter the GUID that identifies a particular Windows Installer .msi package. The Package Code associates an .msi file with an application or product and is represented as a string GUID—a text string that has a special format: {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX} where each X character is a hex digit (0 through 9 or uppercase A through F).
Product Code	Enter the GUID that uniquely identifies the particular product release of the package. The ProductCode is a Windows Installer property and is represented as a string GUID—a text string that has a special format: {XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX} where each X character is a hex digit (0 through 9 or uppercase A through F).

Table 22-5 • Package Attribute Search Fields (cont.)

Metadata	Description
Upgrade Code	<p>Enter the GUID that identifies the family of products that are in the same upgrade path. The UpgradeCode is a Windows Installer property and is represented as a string GUID—a text string that has a special format:</p> <pre>{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXsXXXX}</pre> <p>where each X character is a hex digit (0 through 9 or uppercase A through F).</p>  <p>Note • Each stand-alone product usually has its own UpgradeCode GUID. Every version of XYZ Product typically uses the same GUID for the UpgradeCode. In other words, Product A Version 1.0 has the same UpgradeCode as Product A Version 2.0, but has a different UpgradeCode than Product B.</p>
Setup File Name	Name of the Windows Installer (.msi) file that was imported into the Application Catalog.
Comments	Enter the text of any comments associated with the package.
Import Date	Use this field to search for packages that were imported during a specific date range. Select the greater than or equal (>=) or less than (<) operator from the list, and then click on the box and select a date from the pop-up calendar.
Extended Attributes	Enter the value of any of the Extended Attributes associated with the package.

Package Content

You can search for packages in a catalog based on one or more of any of the following Package Content metadata

Table 22-6 • Package Content Search Fields

Metadata	Description
File	Enter the file name of one of the files in the Windows Installer package.
Registry Key	Enter a registry key to search on.
Registry Value	Enter a registry value to search on.
INI File	Enter any changes to an .ini file that are made when the product is installed.
Shortcut	Enter the name of a shortcut that is created when the product is installed.

Application Request Attributes

You can search for packages in a catalog based on one or more of any of the following attributes of the package's associated Application Request:

Table 22-7 • Application Request Attributes Search Fields

Metadata	Description
Name	Enter the name of the package's associated Application Request.
Upload Date	Enter the date the Application Request was created.
Due Date	Enter the date the Application Request is scheduled to be completed, based upon its value for Application Due Period .
Risk Date	Enter the date at which the Application Request's status will change to At Risk , which is based upon its value for Application At Risk Period .
Due Period	Enter, in days, the length of time this Application Request needs to be completed in order to meet its Project's Service Level Agreement (SLA) requirements.
End Date	Enter the date the Application Request was completed.

Information Included in Package Reports

A Package Report presents information in a tabbed interface. See [Navigating Through a Package Report](#).

A Package Report includes the following major sections:

- [Package Summary Information View](#)
- [Files View](#)
- [Registry View](#)
- [Shortcuts View](#)
- [ODBC Drivers View](#)
- [ODBC DS View](#)
- [Extended Attributes View](#)
- [Validation View](#)
- [Conflicts View](#)
- [History View](#)
- [Dependencies View](#)

Package Summary Information View

The initial view (Page 1) of a Package Report is the **Package Summary Information** view.

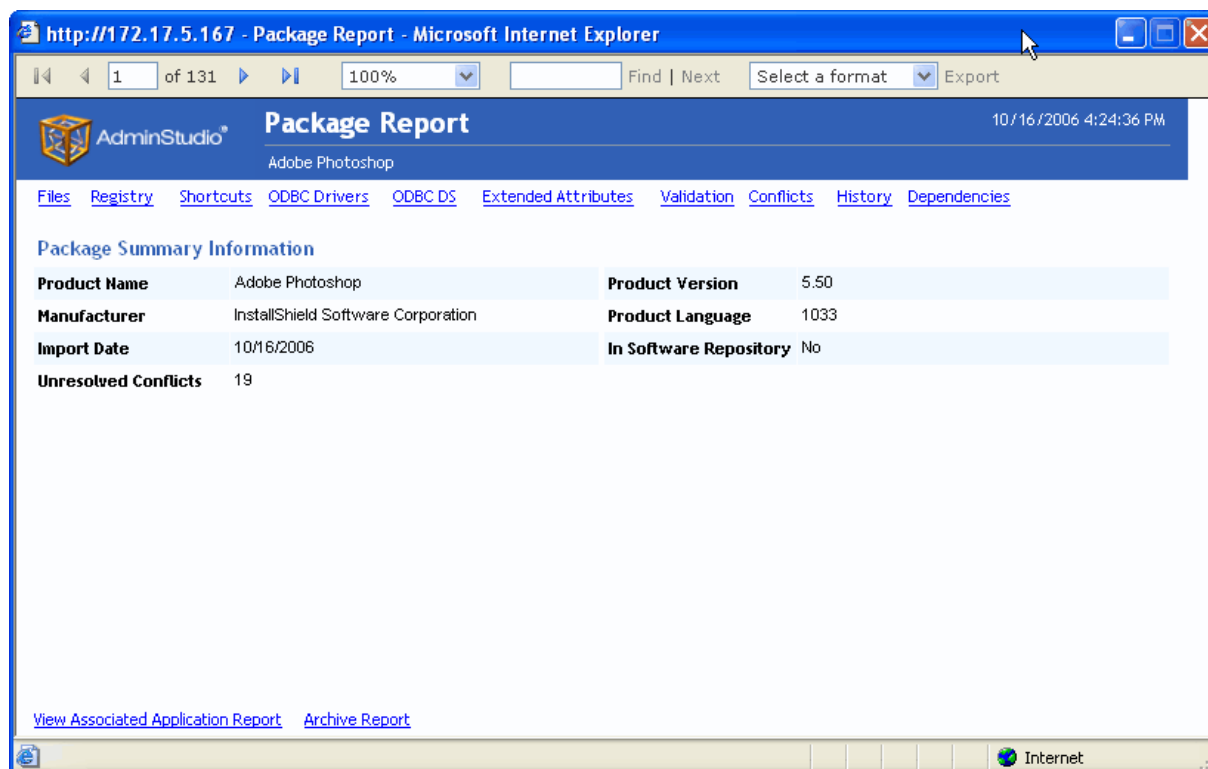


Figure 22-4: Package Report / Package Summary Information View

The Package Summary Information View lists the following information:

Table 22-8 • Package Report / Package Summary Information

Item	Description
Product Name	Name assigned to the package.
Manufacturer	Company that authored the package.
Import Date	The date and time the package was imported into the Application Catalog.
Unresolved Conflicts	The number of detected conflicts, generated during conflict analysis of this package, which have not yet been resolved—either automatically or manually.
Product Version	Version of package that is recorded in the package's Windows Installer file.
Product Language	Decimal-based code identifying the language that this software package was authored for. For example, English is 1033, German is 1031, and Japanese is 1041.

Table 22-8 • Package Report / Package Summary Information (cont.)

Item	Description
In Software Repository	Indicates whether or not this package and its associated files are managed by the Software Repository.

Files View

The **Files** view lists all of the files included in the selected package, and the location where these files will be installed.

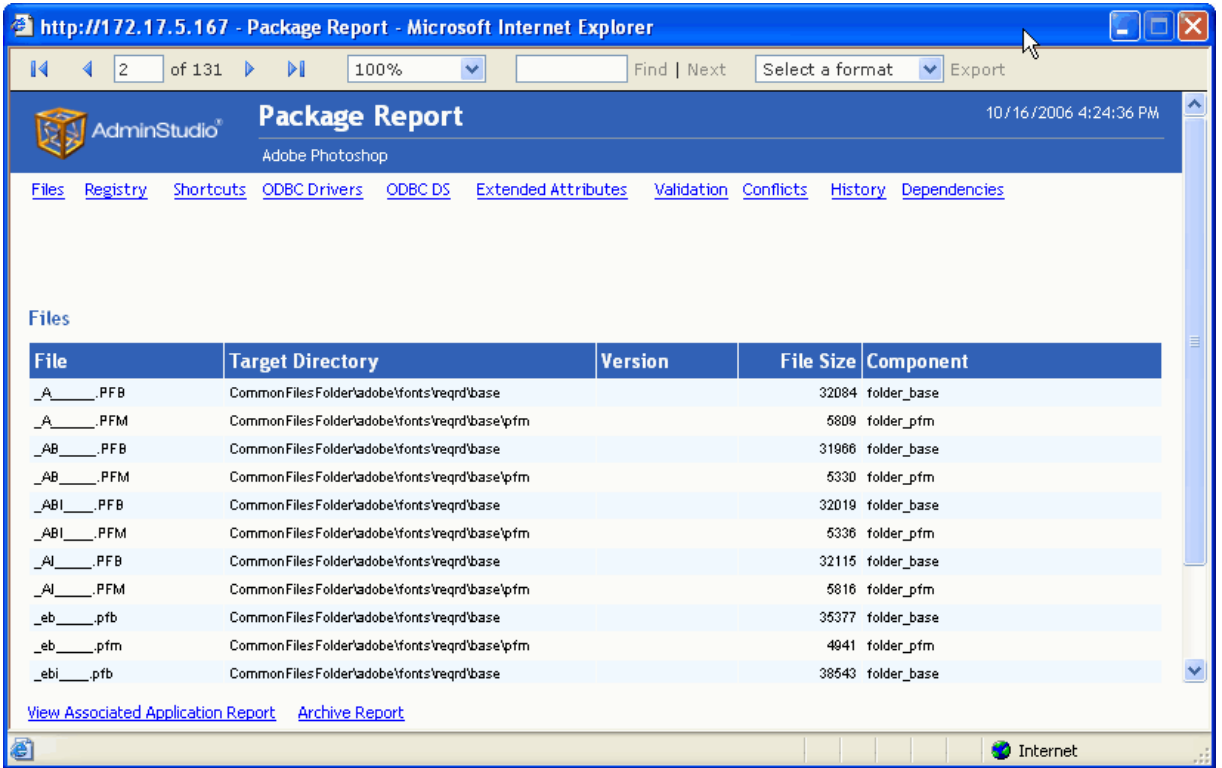


Figure 22-5: Package Report / Files View

For each file, the following information is listed:

Table 22-9 • Package Report / Files Information

Item	Description
File	Name of file included with this package.
Target Directory	Name of directory where the file is installed.
Version	Version number of the file.

Table 22-9 • Package Report / Files Information (cont.)

Item	Description
File Size	Size of the installed file.
Component	Component that the file is associated with.

Registry View

The **Registry** view lists the registry entries that will be created when this package is installed.

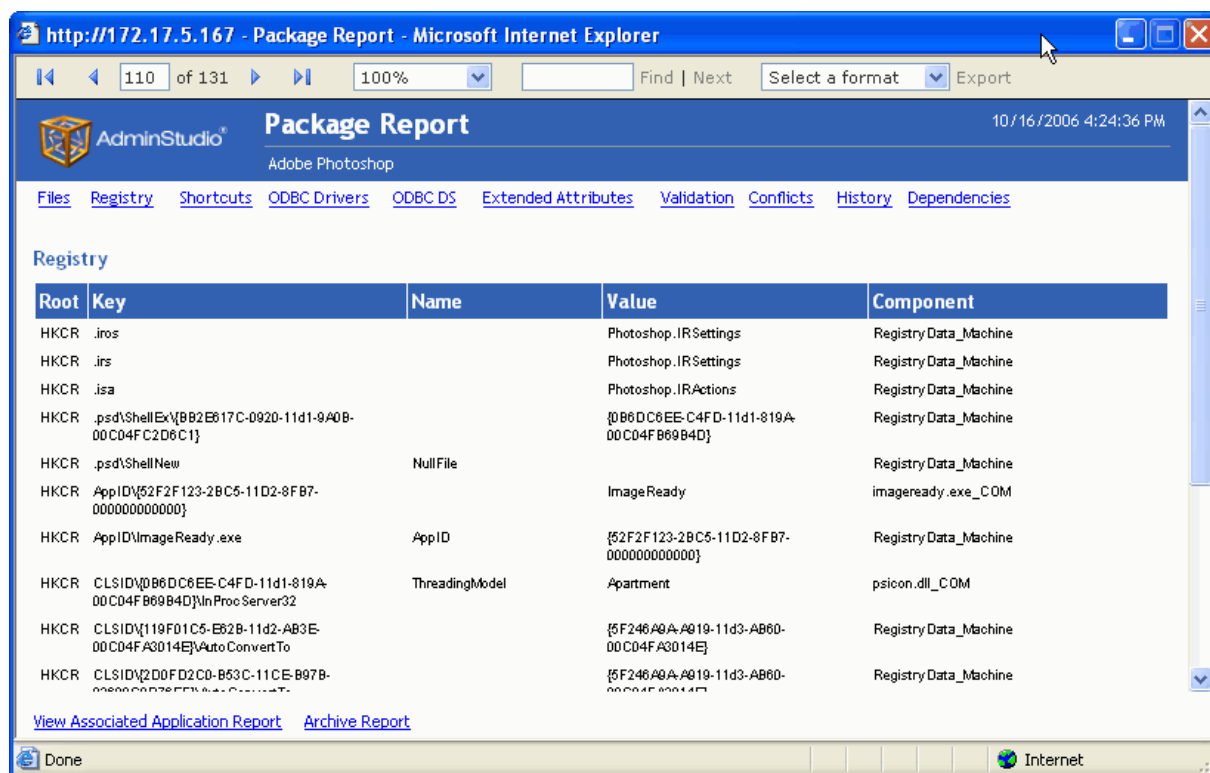


Figure 22-6: Package Report / Registry View

For each registry entry, the following information is listed:

Table 22-10 • Package Report / Registry Information

Item	Description
Root	Identifies the predefined “root” key that contains the registry entry.
Key	A registry key.
Name	Name identifying the registry entry.

Table 22-10 • Package Report / Registry Information (cont.)

Item	Description
Value	The string of data that defines the value of the key.
Component	Package component that the registry entry is associated with.

Shortcuts View

The **Shortcuts** view lists all of the shortcuts that will be created when this package is installed.

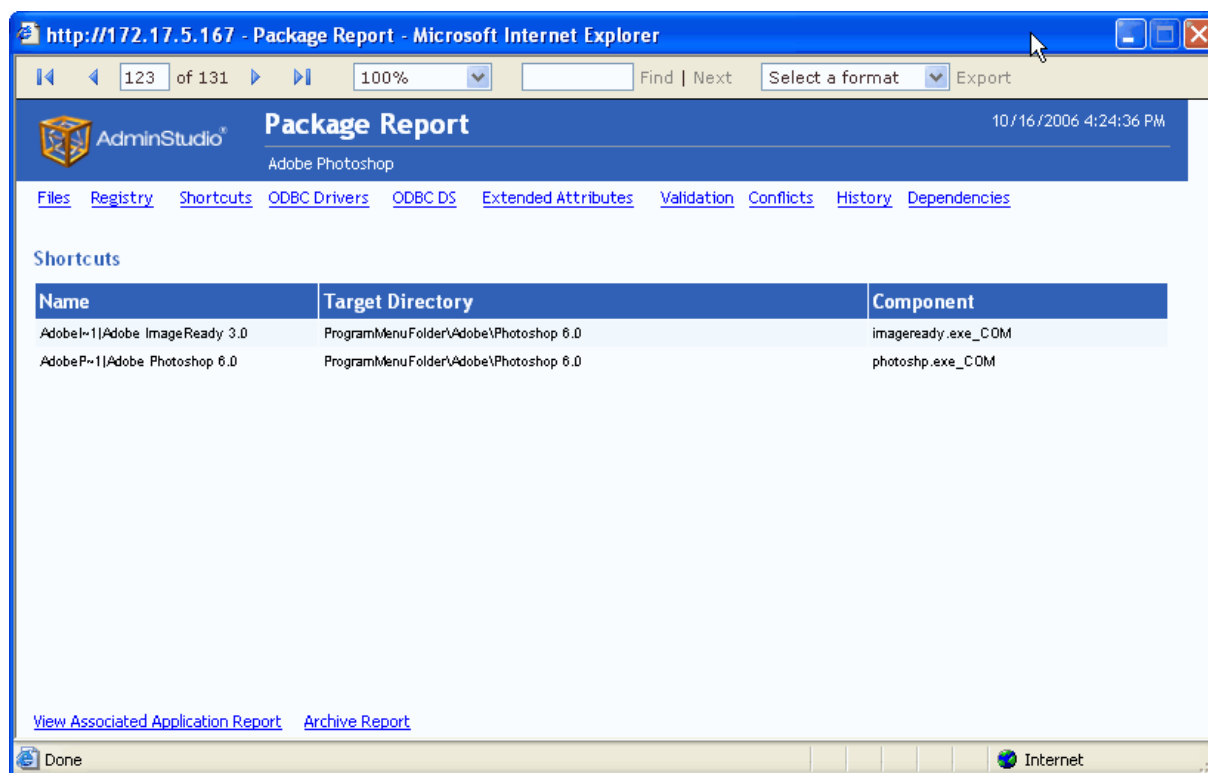


Figure 22-7: Package Report / Shortcuts View

For each shortcut, the following information is listed:

Table 22-11 • Package Report / Shortcuts Information

Item	Description
Name	Name identifying the shortcut.
Target Directory	Directory and executable that the shortcut invokes.
Component	Component associated with the shortcut.

ODBC Drivers View

The **ODBC Drivers** view lists all of the Open Database Connectivity (ODBC) drivers in the package.

ODBC Resources are ones that involve interaction with databases. ODBC drivers are libraries that implement functions involving ODBC. Each database type has its own ODBC driver.

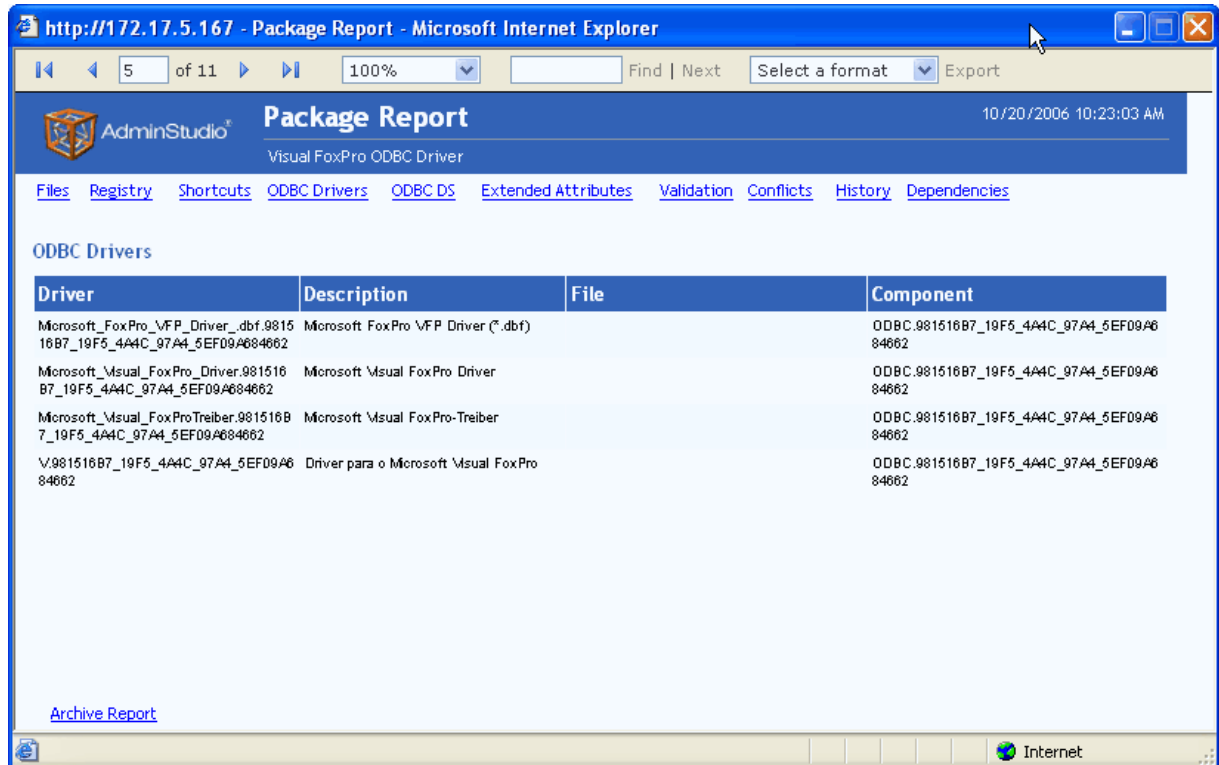


Figure 22-8: Package Report / ODBC Drivers View

For each ODBC driver, the following information is listed:

Table 22-12 • Package Report / ODBC Drivers Information

Item	Description
Driver	Name of an Open Database Connectivity (ODBC) driver in the package. Each database type has its own ODBC driver.
Description	Description of the ODBC driver identifying its associated database type.
File	File associated with the ODBC driver.
Component	Component associated with the ODBC driver.

ODBC DS View

The **ODBC DS** view lists all of the Open Database Connectivity (ODBC) data sources in the package. An ODBC data source identifies the source database type and provides information on how to connect to that database.

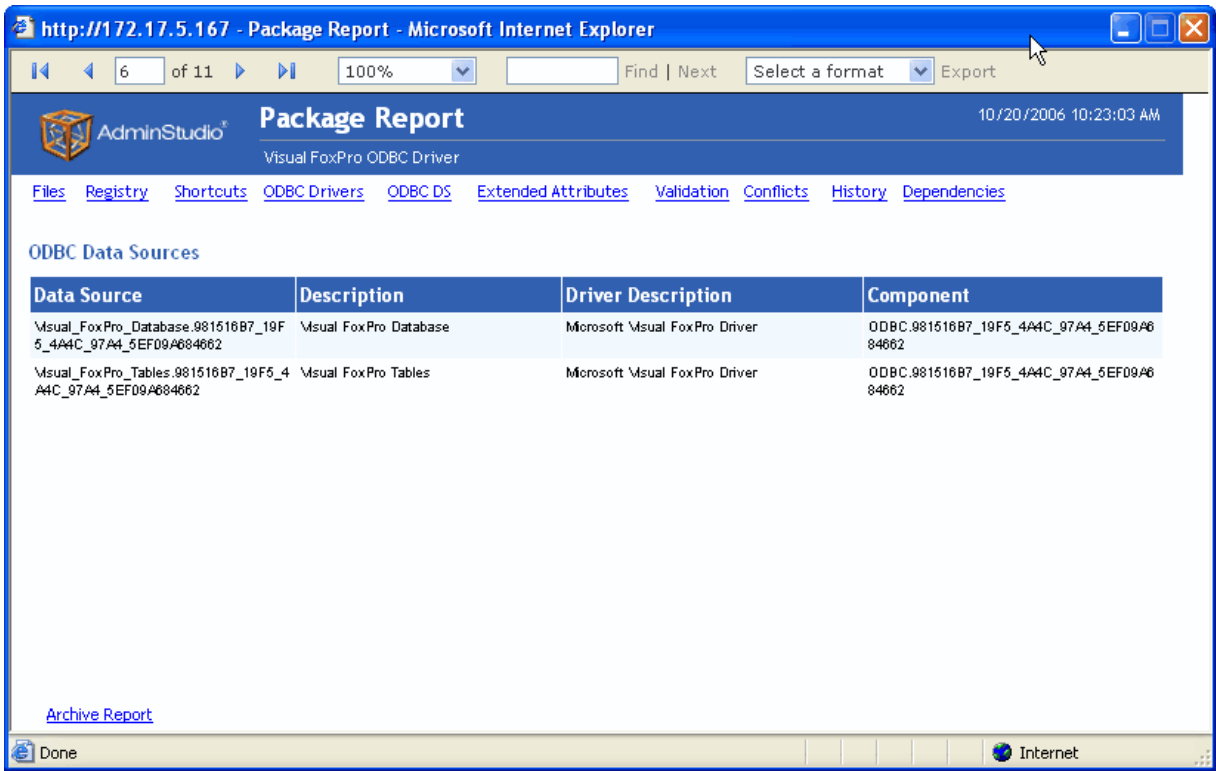


Figure 22-9: Package Report / ODBC Data Sources View

For each ODBC DS, the following information is listed:

Table 22-13 • Package Report / ODBC DS Information

Item	Description
Data Source	Name of the ODBC data source, which identifies the source database type and provides information on how to connect to that database.
Description	Identifies the database type.
Driver Description	Name of this ODBC data source's associated ODBC driver.
Component	Component that this ODBC data source is affiliated with.

Extended Attributes View

The **Extended Attributes** view lists all of the extended attribute metadata that has been entered for this package.

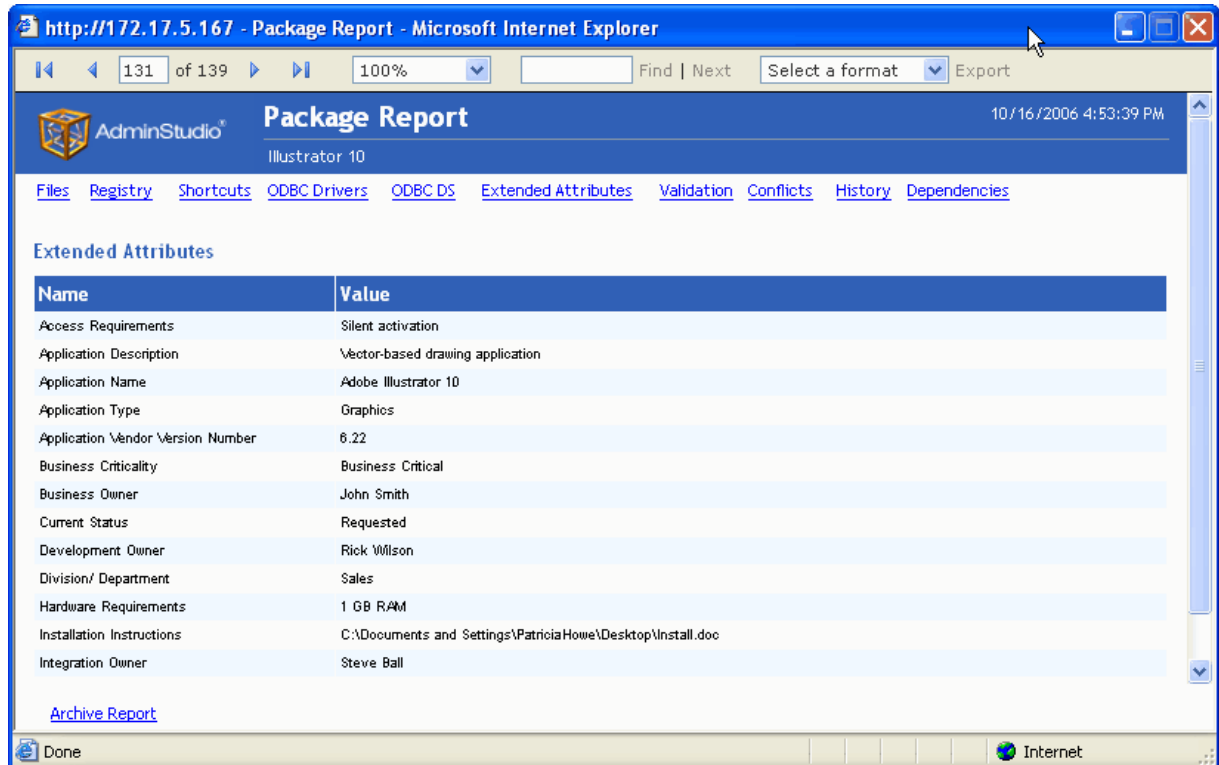


Figure 22-10: Package Report / Extended Attributes View

For each Extended Attribute, the following information is listed:

Table 22-14 • Package Report / Extended Attributes Information

Item	Description
Name	Name identifying the attribute.
Value	Content entered for the attribute.

Validation View

The **Validation** view lists all of the ICE rule errors and warnings that were generated when the package was validated against Microsoft ICEs (Internal Consistency Evaluators)—custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards.

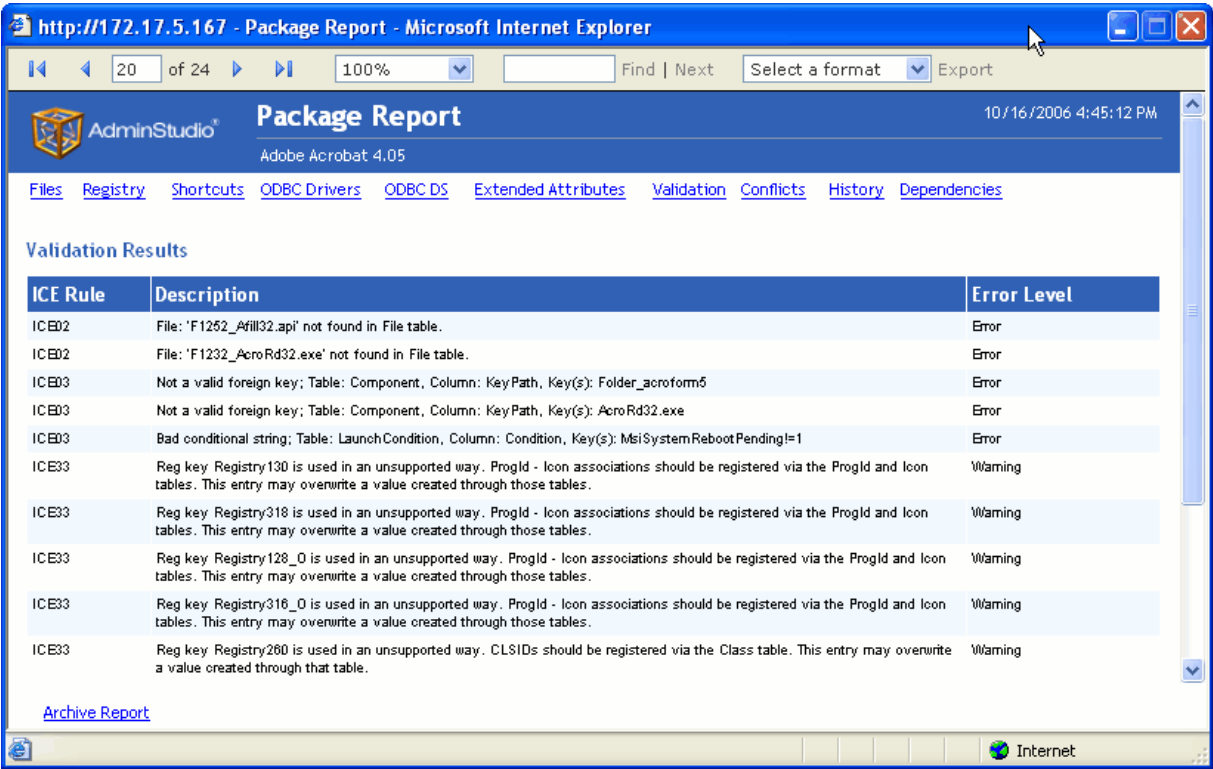


Figure 22-11: Package Report / Validation View

For each error or warning, the following information is listed:

Table 22-15 • Package Report / Validation Information

Item	Description
ICE Rule	Name of ICE Rule that generated an error or warning message.
Description	Error or warning message.
Error Level	Indicates the severity of the message as either being a Warning or an Error. <ul style="list-style-type: none">• Errors—Package authoring that will cause incorrect behavior.• Warnings—Package authoring that could possibly cause incorrect behavior. Warnings can also report unexpected side-effects of package authoring.

Conflicts View

The **Conflicts** view lists all of the unresolved errors that were found when conflict analysis was performed on this package.

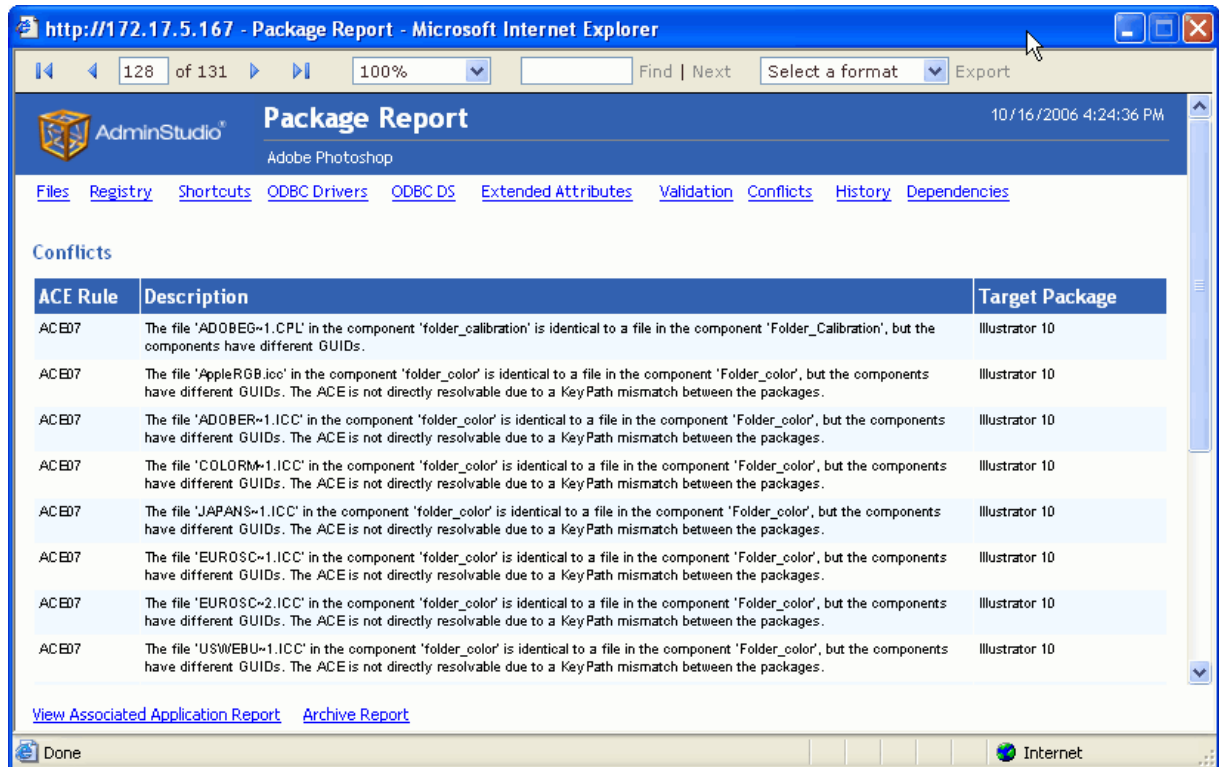


Figure 22-12: Package Report / Conflicts View

For each error, the following information is listed:

Table 22-16 • Package Report / Conflicts Information

Item	Description
ACE Rule	Name of ACE Rule that generated the message.
Description	Message generated during conflict analysis.
Target Package	Package that conflicted with this package.

History View

The **History** view lists all of the actions that have been performed on this package since it was imported into the Application Catalog.

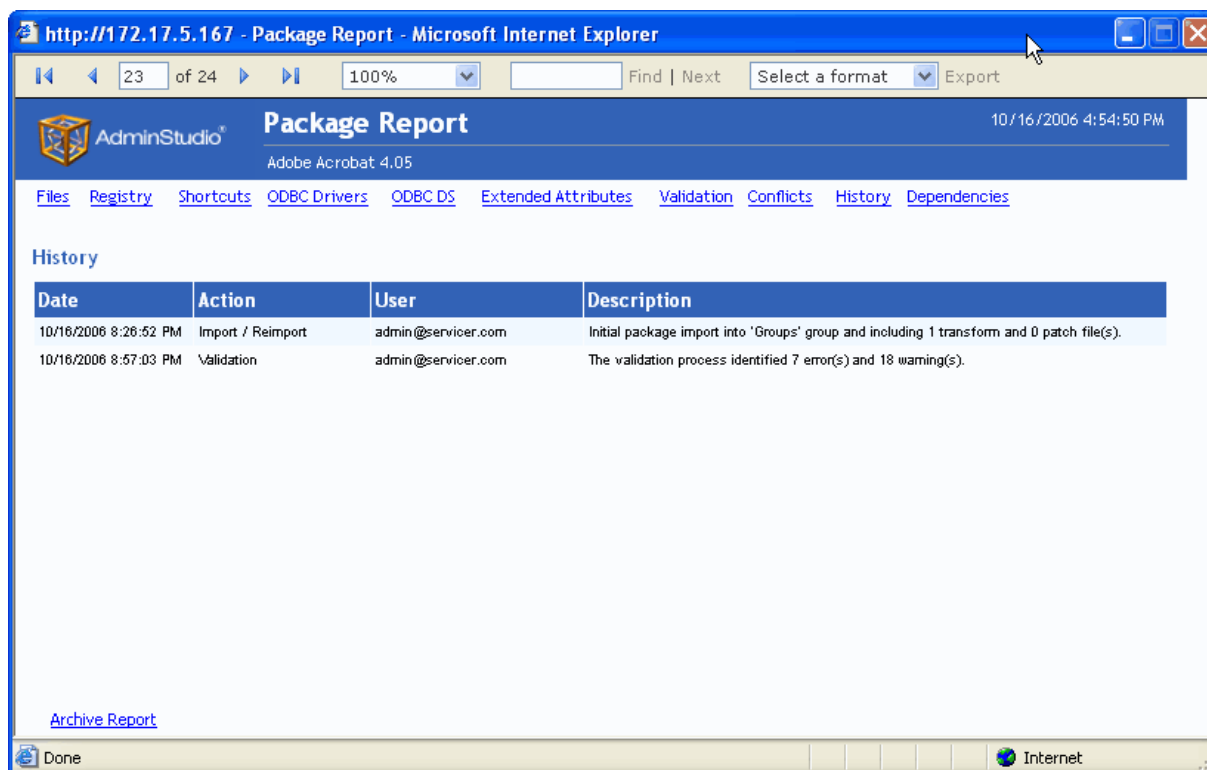


Figure 22-13: Package Report / History View

For each action, the following information is listed:

Table 22-17 • Package Report / History Information

Item	Description
Date	Day and time the event occurred.
Action	Identifies the event that occurred.
User	Identifies the user who executed the event.
Description	Description of the event that occurred.

Dependencies View

The **Dependencies** view lists all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog.

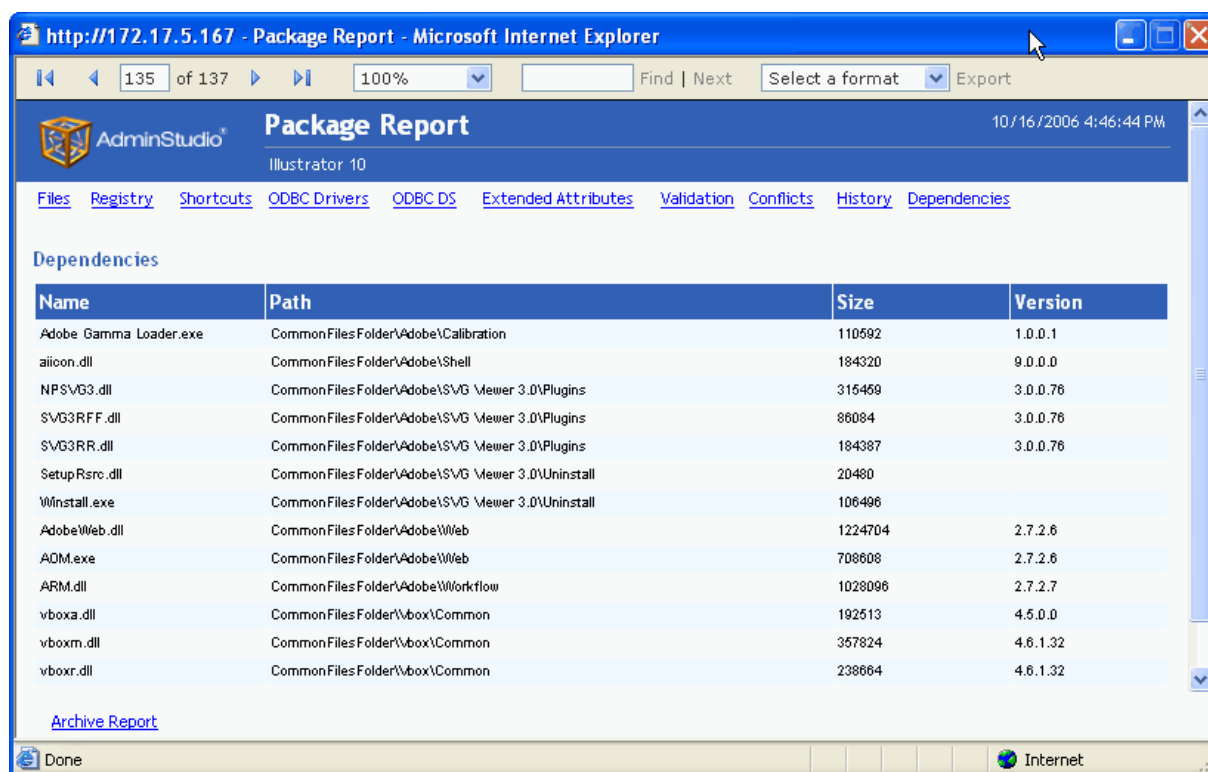


Figure 22-14: Package Report / Dependencies View

For each dependency, the following information is listed:

Table 22-18 • Package Report / Dependencies Information

Item	Description
Name	Name of a file associated with this package that has dependencies with files used by other products or operating systems in the Application Catalog.
Path	Location where this dependent file is installed.
Size	Size of the dependent file.
Version	Version of the dependent file.

Navigating Through a Package Report

The Package Report consists of the initial Package Summary View and 10 other multi-page views which are accessed by clicking the links at the top of the report:

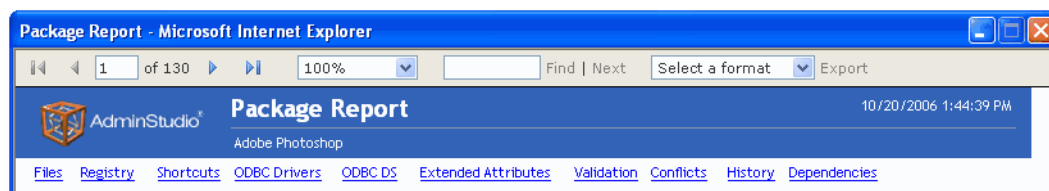


Figure 22-15: Navigation Links on the Package Report

Scrolling Through Pages of a View

Each of the Package Report views can be either a single page or multi-page, depending upon the content. The Package Report window is not resizable, so you cannot enlarge the window to display more items. Instead, you can use the Page Scrolling controls in the toolbar.

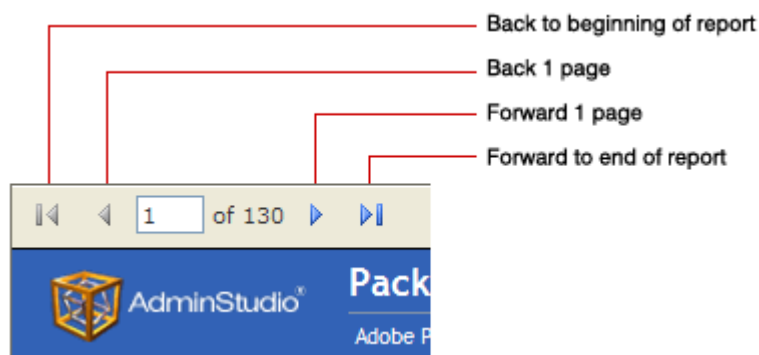


Figure 22-16: Page Scrolling Controls on Package Report

The total number of pages of the Package Report is listed in the toolbar, along with the number of the page that you are currently viewing. To jump to a specific page, enter a number in the box and click **Enter**.

Page 1 of the Package Report is the **Package Summary Information** view. Following this view, the rest of the views follow in the order in which they appear in the navigation links. The total number of pages in a Package Report is determined by adding the number of pages of all of the different views together.

Using Zoom Capability to Modify the Report Size

You can make selections from the Zoom list in the Package Report tool bar to enlarge or decrease the size of the report.

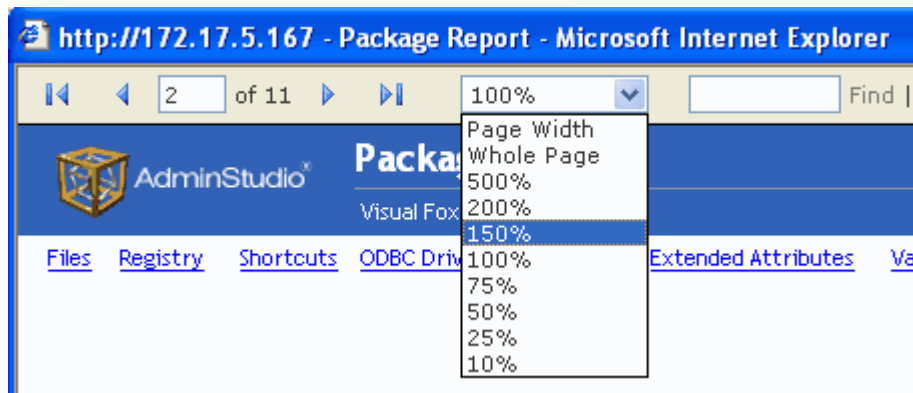


Figure 22-17: Zoom List on the Package Report



Note • When you use the Zoom list to change the size of a Package Report, the size of the font used in the text is increased or decreased; however, the amount of information displayed on one page does not change.

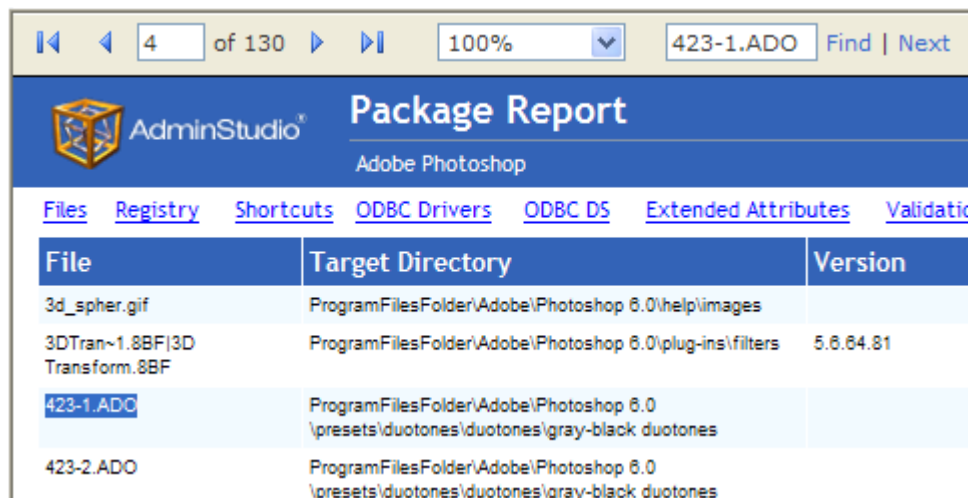
Searching for Information in a Package Report

You can use the **Find** box in the Package Report tool bar to search for specific information in the Package Report.



Task: To search a Package Report:

1. In the Package Report toolbar, enter the text you want to search for in the **Find** box and click **Find**. The page containing the first instance of that text is opened, and the text you searched for is highlighted.



2. Click **Next** in the tool bar to find the next instance of the text.

Archiving a Package Report

You can archive a Package Report to document a snapshot of a package's information as of a specific date and time.

Package Reports are saved in PDF format, and therefore can be easily distributed. An archived report looks very similar to the original report, except that it is a multiple-page PDF:

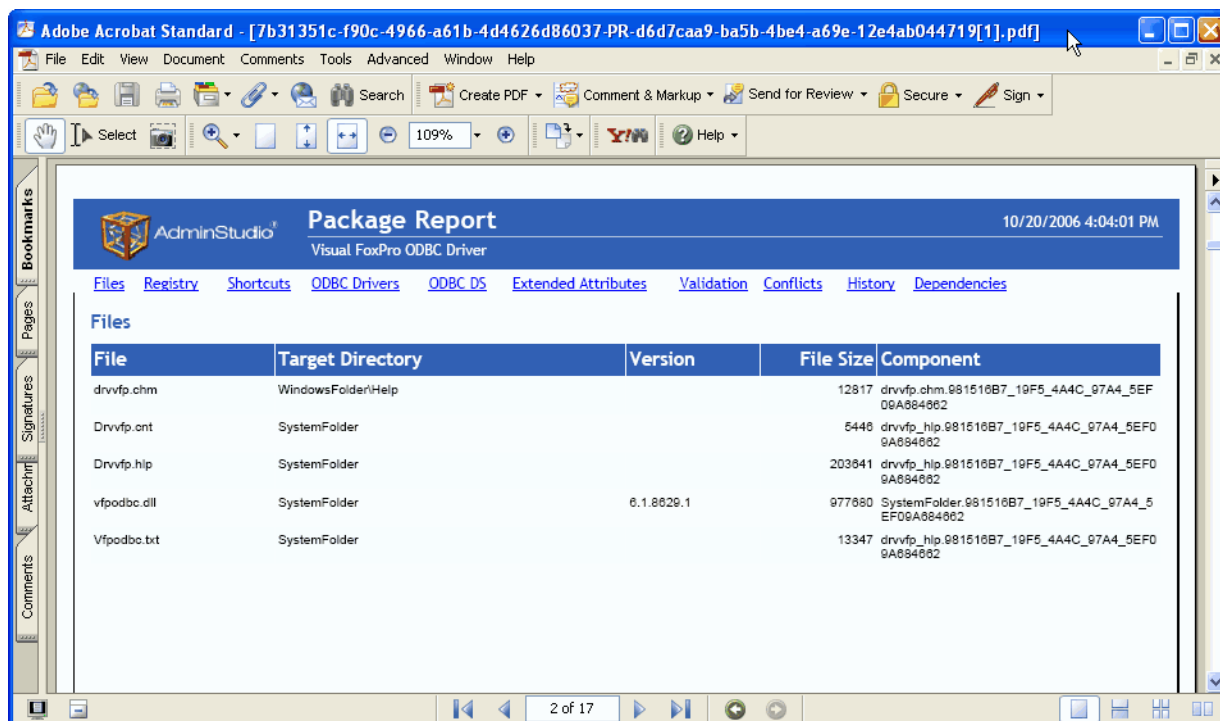


Figure 22-18: Archived Package Report



Note • In an archived Package Report PDF, the navigation links at the top of the report (**Files**, **Registry**, **Shortcuts**, etc.) are not active. To scroll through the PDF, use the standard Adobe Reader controls.

To archive a Package Report, perform the following steps.



Task: **To archive a Package Report:**

1. Open a Package Report.
2. Click the **Archive Report** link in the lower left corner any of the Package Report pages. The report is archived in PDF format and the following message is displayed:

The report has been archived.

3. Click the Report Center **All Reports** tab. The **All Reports** page opens, and the report that you just archived is listed.



Note • Each user's **Archived Reports** list only includes those reports that they archived. If you want others in your organization to view an archived report, you need to distribute the PDF via email or other delivery method.

4. Click **View** next to the Package Report that you want to view. The report is opened in a PDF browser.

Deleting an Archived Package Report from the Archived Reports List

To delete an archived Package Report, perform the following steps.



Task: *To delete an archived Package Report from the Archived Reports list:*

1. In the Archived Reports list on the **All Reports** page, right-click on the archived report you want to delete, and then click **Delete**. You are prompted to confirm the deletion.
2. Click **OK**. The archived report is deleted.

Exporting a Package Report

You can export the contents of a Package Report to an Excel (.xls) or Acrobat (.pdf) file.

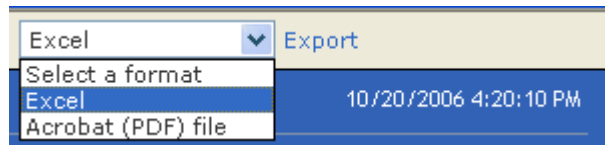
- **Excel .xls file**—When a Package Report is exported to Microsoft Excel format, each of the Package Report views are displayed on a different worksheet.
- **Acrobat .pdf file**—An exported Package Report in PDF format is the same as the PDF created when a Package Report is archived. See [Archiving a Package Report](#).

To export a Package Report, perform the following steps.

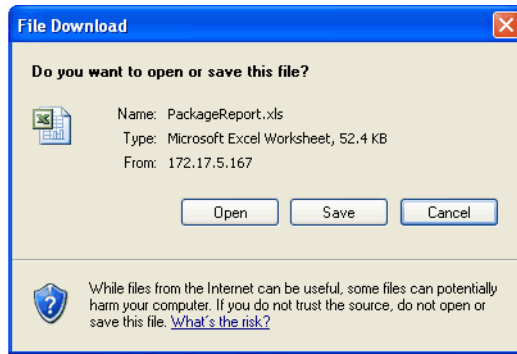


Task: *To export a Package Report:*

1. In the Package Report tool bar, select **Excel** or **Acrobat (PDF) file** from the list.



2. Click **Export**. The File Download dialog box opens.



3. Click **Save**. The **Save As** dialog box opens.
4. Select a location for the exported file and click **Save**.

Generating a Custom SQL Query Report for AdminStudio

You can generate a Custom SQL Query Report to include data generated by both AdminStudio and Workflow Manager. To generate a Custom SQL Query Report, perform the following steps.



Task: *To generate a new Custom SQL Query report:*

1. Open the **Report Center** tab. The **All Reports** page opens.
2. Click **Add**. The **Select Report Type** page opens.
3. Click **Custom SQL Query Report**. The **Enter SQL Query Panel** opens.
4. Enter an SQL Query to retrieve the data for this report. Click the **Test Query** button to verify the query syntax.



Tip • To assist you in writing queries to retrieve data, see [AdminStudio Tables](#), for information on each table and the columns it contains, and [Wildcard Support in Report Center SQL Queries](#).

5. Click **Next**. The **General Information Panel** opens.
6. Enter a **Report Name** and **Description** to clearly identify the contents and purpose of this report. This name and description will be listed on the **All Reports** page.
7. Select the Roles that you want to have permission to view this report.
8. Click **Next**. The **Summary Panel** opens, which displays all the information needed to create the report.
9. Click **Finish**. The report is generated in the **Report View**. This report is also saved and now appears in the list on the **All Reports** page.

Wildcard Support in Report Center SQL Queries

In Report Center searches, the LIKE operator is always used. You can combine the LIKE operator with a wildcard character, and the following rules apply:

Table 22-19 • Wildcard Support in Report Center Queries

Situation	Rule
When no wildcards are used	<p>If you do not enter a wildcard character in the Search box, then Report Center performs a “LIKE” search, which searches for any occurrence of that text anywhere in the item that is being searched for.</p> <p>For example, if you are searching for a file name that has the word test anywhere in the file name, and you entered test in the Search box, it would be interpreted by Report Center as:</p> <p><code>*test*</code></p> <p>And the following files would be found:</p> <p>MyTestFile and TestFile</p>
When wildcards are used	<p>You can specify a * wildcard in the Search box to narrow the search results.</p> <p>For example, if you are searching for a file name that includes the word test, but does not begin with it, and you entered <code>*test</code> in the Search box, MyTest would be returned, but not TestFile.</p>

Report Center Reference

This section includes reference information on the following pages, views, and reports:

- [All Reports Page](#)
- [Report View](#)
- [Select Package Page](#)
- [Package Report](#)
- [Reports Wizard](#)

All Reports Page

The **All Reports** page provides access to the archived Package Reports and Custom SQL Query reports.

The All Reports page includes the following options:

Option	Description
Report Name	Name of a saved report.
Report Description	Text describing the contents of the report.
Add	Click to design and generate a new report.

Available Reports

From the **All Reports** page, you can choose to create a new Custom SQL Query Report, or view a report that was already created. From this page you can also open an archived Package Report.

Viewing an Existing Custom Report from the All Reports Page

To view a report, click **View** to launch the **Report View**.

Report View

The Report View, which is opened by clicking **View** next to an existing Custom Report on the **All Reports** page, displays the contents of a saved report.

Table 22-20 • Report View

Option	Description
Report Name	Name of this report.
Description	Text describing the contents of the report.
Default Filter	Name of filter, if any, applied to this report.
Export	Click to export this report to Microsoft Excel format.

Select Package Page

From the **Select Package** page, you can select or search for a specific package, and then generate a detailed Package Report. Information about the **Select Package** page is presented in the following sections:

- [Areas of the Select Package Page](#)
- [Group Summary Information](#)
- [Package Summary Information](#)
- [Search Package Area](#)

Areas of the Select Package Page

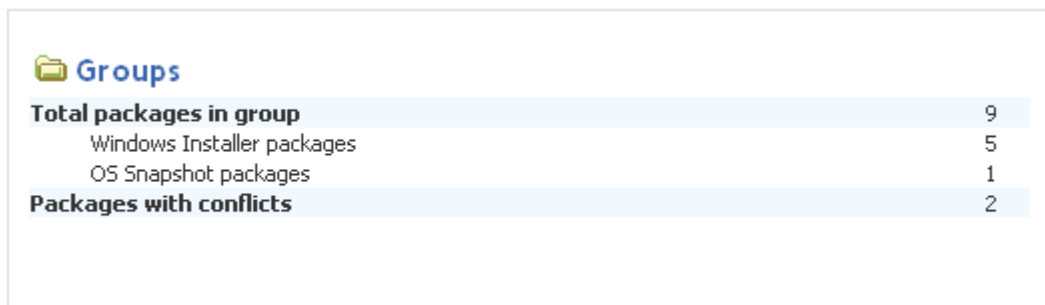
The **Select Package** page is organized in the following areas:

Table 22-21 • Areas on the Select Package Page

Area	Description
Package Tree	On the left side of the interface, all of the groups and packages in the AdminStudio Enterprise Server Application Catalog are displayed in a tree control. If a search has been applied using the fields in the Search Packages area, only those packages meeting that criteria are listed, and the Show All packages button is enabled.
Package or Group Summary Area	In the top half of the right side of the interface, either a summary of group statistics (when a group is selected) or a summary of package statistics (when a package or OS Snapshot is selected) is displayed. See Group Summary Information and Package Summary Information . When a package is selected, you can click View Report to view that package's Package Report.
Search Packages Area	In bottom half of the right side of the interface, the Search Packages area includes fields allowing you to search for packages by attributes, content, or associated Application Requests.

Group Summary Information

When a group in the package tree is selected, summary information about that group is listed.



The screenshot shows a summary for a group named "Groups". It lists the total packages in the group as 9, broken down into 5 Windows Installer packages and 1 OS Snapshot package. It also shows that there are 2 packages with conflicts.


 Groups	
Total packages in group	9
Windows Installer packages	5
OS Snapshot packages	1
Packages with conflicts	2

Figure 22-19: Group Summary on the Select Package Page

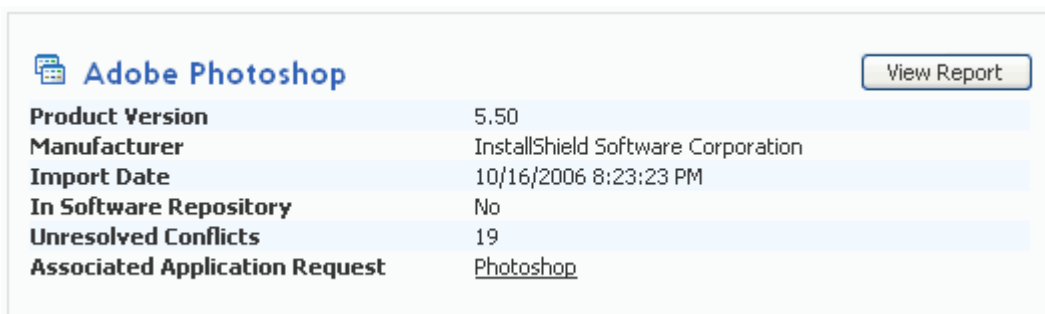
The following information about the selected group is listed:

Table 22-22 • Group Summary Information

Item	Description
Total Packages in Group	Total number of packages that have been imported into the selected group. This number is also broken down between number of Windows Installer packages and the number of OS Snapshot packages.
Packages With Conflicts	Number of packages that have unresolved conflicts with other packages or OS Snapshots in the Application Catalog.

Package Summary Information

When a package in the package tree is selected, summary information about that package is listed.



The screenshot shows a summary for the "Adobe Photoshop" package. It lists the product version as 5.50, the manufacturer as InstallShield Software Corporation, the import date as 10/16/2006 8:23:23 PM, and that it is not in the software repository. It also shows 19 unresolved conflicts and an associated application request for "Photoshop". A "View Report" button is visible in the top right corner.


 Adobe Photoshop View Report	
Product Version	5.50
Manufacturer	InstallShield Software Corporation
Import Date	10/16/2006 8:23:23 PM
In Software Repository	No
Unresolved Conflicts	19
Associated Application Request	Photoshop

Figure 22-20: Package Summary on the Select Package Page

The following information about the selected group is listed:

Table 22-23 • Package Summary Information

Item	Description
Product Version	Version of package that is recorded in the package's Windows Installer file.
Manufacturer	Company that authored the package.
Import Date	The date and time the package was imported into the Application Catalog.
In Software Repository	Indicates whether or not this package and its associated files are managed by the Software Repository.
Unresolved Conflicts	The number of detected conflicts, generated during conflict analysis of this package, which have not yet been resolved—either automatically or manually.
Associated Application Request	If this package is associated with an Application Request, the Application Request name is listed here. Click this link to open the Application Request's Application Report.

Search Package Area

In the **Search Package** area of the **Select Package** page, you can filter the list of packages displayed in the package tree to display only those packages that meet specific search criteria, which are grouped into three categories:

- **Package Attributes**—Search by properties assigned to the Windows Installer package. See [Package Attributes](#).
- **Package Content**—Search by files, registry entries, .ini files, or shortcuts contained in the Windows Installer package. See [Package Content](#).
- **Application Request Attributes**—Search by information related to a package's associated Application Request. See [Application Request Attributes](#).

To filter the list of packages displayed in the package tree to display only those packages that meet specific search criteria, enter values in the criteria fields that you want to search on, and click **Search**. The packages that meet *any of the criteria* are then listed in the package tree in alphabetical order and are no longer grouped.

Package Attributes

You can search for packages in a catalog based on one or more of any of the following Package attribute metadata:

Table 22-24 • Package Attribute Search Fields

Metadata	Description
Product Name	Enter a package name to search by.

Table 22-24 • Package Attribute Search Fields (cont.)

Metadata	Description
Manufacturer	Enter the name of the software vendor who authored the package.
Product Version	Enter the value that was entered in the Windows Installer ProductVersion property for this package.
Package Code	<p>Enter the GUID that identifies a particular Windows Installer .msi package. The Package Code associates an .msi file with an application or product and is represented as a string GUID—a text string that has a special format:</p> <p>{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}</p> <p>where each X character is a hex digit (0 through 9 or uppercase A through F).</p>
Product Code	<p>Enter the GUID that uniquely identifies the particular product release of the package. The ProductCode is a Windows Installer property and is represented as a string GUID—a text string that has a special format:</p> <p>{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX}</p> <p>where each X character is a hex digit (0 through 9 or uppercase A through F).</p>
Upgrade Code	<p>Enter the GUID that identifies the family of products that are in the same upgrade path. The UpgradeCode is a Windows Installer property and is represented as a string GUID—a text string that has a special format:</p> <p>{XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXsXXXX}</p> <p>where each X character is a hex digit (0 through 9 or uppercase A through F).</p> <div data-bbox="537 1197 574 1241" data-label="Image"> </div> <p>Note • Each stand-alone product usually has its own UpgradeCode GUID. Every version of XYZ Product typically uses the same GUID for the UpgradeCode. In other words, Product A Version 1.0 has the same UpgradeCode as Product A Version 2.0, but has a different UpgradeCode than Product B.</p>
Setup File Name	Name of the Windows Installer (.msi) file that was imported into the Application Catalog.
Comments	Enter the text of any comments associated with the package.
Import Date	Use this field to search for packages that were imported during a specific date range. Select the greater than or equal (\geq) or less than (\leq) operator from the list, and then click on the box and select a date from the pop-up calendar.
Extended Attributes	Enter the value of any of the Extended Attributes associated with the package.

Package Content

You can search for packages in a catalog based on one or more of any of the following Package Content metadata

Table 22-25 • Package Content Search Fields

Metadata	Description
File	Enter the file name of one of the files in the Windows Installer package.
Registry Key	Enter a registry key to search on.
Registry Value	Enter a registry value to search on.
INI File	Enter any changes to an .ini file that are made when the product is installed.
Shortcut	Enter any shortcuts that are created when the product is installed.

Application Request Attributes

You can search for packages in a catalog based on one or more of any of the following attributes of the package's associated Application Request:

Table 22-26 • Application Request Attributes Search Fields

Metadata	Description
Name	Enter the name of the package's associated Application Request.
Upload Date	Date the Application Request was submitted.
Due Date	Enter the date the Application Request is scheduled to be completed, based upon its value for Application Due Period .
Risk Date	Enter the date at which the Application Request's status will change to At Risk , which is based upon its value for Application At Risk Period .
Due Period	Enter, in days, the length of time this Application Request needs to be completed in order to meet its Project's Service Level Agreement (SLA) requirements.
End Date	Enter the date the Application Request was completed.

Package Report

This topic lists the data that is displayed in each section of a Package Report.

- [Package Summary Information View](#)
- [Files View](#)
- [Registry View](#)
- [Shortcuts View](#)
- [ODBC Drivers View](#)
- [ODBC DS View](#)
- [Extended Attributes View](#)
- [Validation View](#)
- [Conflicts View](#)
- [History View](#)
- [Dependencies View](#)



Note • See also see [Information Included in Package Reports](#).

Package Summary Information View

The initial view (Page 1) of a Package Report is the **Package Summary Information** view, and it lists the following information:

Table 22-27 • Package Report / Package Summary Information

Item	Description
Product Name	Name assigned to the package.
Manufacturer	Company that authored the package.
Import Date	The date and time the package was imported into the Application Catalog.
Unresolved Conflicts	The number of detected conflicts, generated during conflict analysis of this package, which have not yet been resolved—either automatically or manually.
Product Version	Version of package that is recorded in the package's Windows Installer file.
Product Language	Decimal-based code identifying the language that this software package was authored for. For example, English is 1033, German is 1031, and Japanese is 1041.

Table 22-27 • Package Report / Package Summary Information (cont.)

Item	Description
In Software Repository	Indicates whether or not this package and its associated files are managed by the Software Repository.

Files View

The **Files** view lists the all of the files included in the selected package, and the location where these files will be installed. For each file, the following information is listed:

Table 22-28 • Package Report / Files Information

Item	Description
File	Name of file included with this package.
Target Directory	Name of directory where the file is installed.
Version	Version number of the file.
File Size	Size of the installed file.
Component	Component that the file is associated with.

Registry View

The **Registry** view lists the registry entries that will be created when this package is installed. For each registry entry, the following information is listed:

Table 22-29 • Package Report / Registry Information

Item	Description
Root	Identifies the predefined “root” key that contains the registry entry.
Key	A registry key.
Name	Name identifying the registry entry.
Value	The string of data that defines the value of the key.
Component	Package component that the registry entry is associated with.

Shortcuts View

The **Shortcuts** view lists all of the shortcuts that will be created when this package is installed. For each shortcut, the following information is listed:

Table 22-30 • Package Report / Shortcuts Information

Item	Description
Name	Name identifying the shortcut.
Target Directory	Directory and executable that the shortcut invokes.
Component	Component associated with the shortcut.

ODBC Drivers View

The **ODBC Drivers** view lists all of the Open Database Connectivity (ODBC) drivers in the package.

ODBC Resources are ones that involve interaction with databases. ODBC drivers are libraries that implement functions involving ODBC. Each database type has its own ODBC driver. For each ODBC driver, the following information is listed:

Table 22-31 • Package Report / ODBC Drivers Information

Item	Description
Driver	Name of an Open Database Connectivity (ODBC) driver in the package. Each database type has its own ODBC driver.
Description	Description of the ODBC driver identifying its associated database type.
File	File associated with the ODBC driver.
Component	Component associated with the ODBC driver.

ODBC DS View

The **ODBC DS** view lists all of the Open Database Connectivity (ODBC) data sources in the package. An ODBC data source identifies the source database type and provides information on how to connect to that database. For each ODBC DS, the following information is listed:

Table 22-32 • Package Report / ODBC DS Information

Item	Description
Data Source	Name of the ODBC data source, which identifies the source database type and provides information on how to connect to that database.
Description	Identifies the database type.

Table 22-32 • Package Report / ODBC DS Information (cont.)

Item	Description
Driver Description	Name of this ODBC data source's associated ODBC driver.
Component	Component that this ODBC data source is affiliated with.

Extended Attributes View

The **Extended Attributes** view lists all of the extended attribute metadata that has been entered for this package. For each Extended Attribute, the following information is listed:

Table 22-33 • Package Report / Extended Attributes Information

Item	Description
Name	Name identifying the attribute.
Value	Content entered for the attribute.

Validation View

The **Validation** view lists all of the ICE rule errors and warnings that were generated when the package was validated against Microsoft ICEs (Internal Consistency Evaluators)—custom actions written by Microsoft which can be executed to determine if an installation package is built according to Windows Installer standards.

For each error or warning, the following information is listed:

Table 22-34 • Package Report / Validation Information

Item	Description
ICE Rule	Name of ICE Rule that generated an error or warning message.
Description	Error or warning message.
Error Level	Indicates the severity of the message as either being a Warning or an Error. <ul style="list-style-type: none"> • Errors—Package authoring that will cause incorrect behavior. • Warnings—Package authoring that could possibly cause incorrect behavior. Warnings can also report unexpected side-effects of package authoring.

Conflicts View

The **Conflicts** view lists all of the unresolved errors that were found when conflict analysis was performed on this package. For each error, the following information is listed:

Table 22-35 • Package Report / Conflicts Information

Item	Description
ACE Rule	Name of ACE Rule that generated the message.
Description	Message generated during conflict analysis.
Target Package	Package that conflicted with this package.

History View

The **History** view lists all of the actions that have been performed on this package since it was imported into the Application Catalog. For each action, the following information is listed:

Table 22-36 • Package Report / History Information

Item	Description
Date	Day and time the event occurred.
Action	Identifies the event that occurred.
User	Identifies the user who executed the event.
Description	Description of the event that occurred.

Dependencies View

The **Dependencies** view lists all of a package's files that have dependencies with files used by other products or operating systems in the Application Catalog. For each dependency, the following information is listed:

Table 22-37 • Package Report / Dependencies Information

Item	Description
Name	Name of a file associated with this package that has dependencies with files used by other products or operating systems in the Application Catalog.
Path	Location where this dependent file is installed.
Size	Size of the dependent file.
Version	Version of the dependent file.

Reports Wizard

Using the Reports Wizard, you can generate a Custom SQL Query report. See [Generating a Custom SQL Query Report for AdminStudio](#) and the [Enter SQL Query Panel](#).

Select Report Type Panel

On the **Select Report Type Panel** of the Reports Wizard, select one of the following options:

Table 22-38 • Select Report Type Panel Options

Option	Description
Custom Report	Select to create a custom report.
Activity Report	Select to create report that exclusively lists Application Request activities.
Custom SQL Query Report	Select to create a custom report defined by entering an SQL query in the Report Wizard.

Enter SQL Query Panel

On the **Enter SQL Query Panel** of the **Custom SQL Query Reports Wizard**, enter an SQL query to retrieve the data for this report. Click the **Test Query** button to verify the query syntax, and click **Next** to proceed.

Wildcard Support in Report Center SQL Queries

In Report Center searches, the LIKE operator is always used. You can combine the LIKE operator with a wildcard character, and the following rules apply:

Table 22-39 • Wildcard Support in Report Center Queries

Situation	Rule
When no wildcards are used	<p>If you do not enter a wildcard character in the Search box, then Report Center performs a “LIKE” search, which searches for any occurrence of that text anywhere in the item that is being searched for.</p> <p>For example, if you are searching for a file name that has the word test anywhere in the file name, and you entered test in the Search box, it would be interpreted by Report Center as:</p> <p>*test*</p> <p>And the following files would be found:</p> <p>MyTestFile and TestFile</p>

Table 22-39 • Wildcard Support in Report Center Queries

Situation	Rule
When wildcards are used	You can specify a * wildcard in the Search box to narrow the search results. For example, if you are searching for a file name that includes the word test, but does not begin with it, and you entered *test in the Search box, MyTest would be returned, but not TestFile.



Tip • To assist you in writing queries to retrieve data, see [AdminStudio and Workflow Manager Tables](#) for information on each table and the columns it contains.

General Information Panel

On the **General Information Panel** of the Reports Wizard, enter a **Report Name** and **Description** to clearly identify the contents and purpose of this report. This name and description will be listed on the **All Reports** page.

Next, select the **Roles** that you want to have permission to view this report.

Click **Next** to continue with the Reports Wizard.



Note • You can change the selected Roles at any time after this report is created by clicking *Edit* next to the Report Name on the **All Reports** page.

Summary Panel

On the **Summary Panel** of the Reports Wizard, the following information is listed:

Table 22-40 • Summary Panel of the Reports Wizard

Option	Description
Report Name	Name of report.
Report Fields	List of fields that you selected to be included in this report.
Template Data	A list of the Template data you selected to be in this report.
Filters	A list of filters applied to this report.

Click **Finish** to generate the Report. The report is generated in the **Report View**. This report is also saved and now appears in the list on the **All Reports** page.

AdminStudio and Workflow Manager Tables

To assist you in writing queries for Custom SQL Query reports, information on AdminStudio and Workflow Manager tables is provided.

- [AdminStudio Tables](#)
- [Workflow Manager Tables](#)

AdminStudio Tables

To assist you in writing queries to retrieve AdminStudio data, information on the following AdminStudio tables is provided.

- [cstblPackage](#)
- [cstblGroups](#)
- [cstblGroupPackages](#)
- [AsJob](#)
- [AsJobQueue](#)
- [AsJobStep](#)

cstblPackage

The cstblPackage table is the core table that contains information about imported packages. This table contains a unique Package ID (the RowID column) that is referenced by all other data tables. This table contains the following columns:

Table 22-41 • cstblPackage Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
PackageCode	text		N	The value of the equivalent MSI PackageCode. Code can be up to 100 characters long.
ProductCode	text		N	The value of the equivalent MSI ProductCode. Code can be up to 100 characters long.
ProductVersion	text		N	The value of the equivalent MSI ProductVersion. Version can be up to 100 characters long.

Table 22-41 • cstblPackage Columns

Column	Type	Key	Nullable	Description
UpgradeCode	text		N	The value of the equivalent MSI UpgradeCode. Code can be up to 100 characters long.
ProductName	text		N	The value of the equivalent MSI ProductName. Name can be up to 255 characters long.
Manufacturer	text		N	The value of the equivalent MSI Manufacturer. Name can be up to 255 characters long.
ProductLanguage	int		N	The value of the equivalent MSI ProductLanguage.
FileName	text		N	UNC path to the MSI package. Name can be up to 255 characters long.
Comments	text		N	User description of the package. Comments can be up to 4000 characters long.
ImportDate	datetime		N	Date package was imported.
FileModifiedDate	datetime		N	Derived from the file system, this is the date that the package (MSI) file was last modified.
Flags	int		N	Used internally.
DisplayedProductName	text		N	Name of package that is displayed in the Application Manager and ConflictSolver Products View. Name can be up to 255 characters long.

cstblGroups

Application Manager allows categorizing related packages using Groups. The cstblGroups table is used to store the information about Application Manager groups. Application Manager ships with one row in this table that defines the root node in the Groups tree. A group may contain imported packages or other sub-groups; these parent-child relationship details are stored in the [cstblGroupPackages](#) table. This table contains the following columns:

Table 22-42 • cstblGroups Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
GroupName	text		N	Name of group. Name can be up to 72 characters long.
Description	text		N	Description of group. Description can be up to 128 characters long.
Comments	text		N	Comments associated with group. Comments can be up to 255 characters long.
Properties	int		N	Used internally.
CreatedDate	datetime		N	Date group was created.
ModifiedDate	datetime		N	Date group was last modified.

cstblGroupPackages

This table defines the relationship between packages and groups. This table contains the following columns:

Table 22-43 • cstblGroupPackages Columns

Column	Type	Key	Nullable	Description
RowID	int	Y	N	The Primary Key—used to uniquely identify a record.
ParentGroupCode	int		N	Foreign Key identifier into the cstblGroup table. Parent group containing either the package or the group entry (Group Row ID).
PkgRowID_	int		N	Foreign Key identifier into the cstblPackage table. If this is a group, the value in this column is 0. Otherwise, it contains the PackageRowID.

Table 22-43 • cstblGroupPackages Columns

Column	Type	Key	Nullable	Description
MemberGroupCode	int		N	Foreign Key identifier into the cstblGroup table. If this is a package, the value in this column is -1. Otherwise, it contains the GroupRowID.

AsJob

This table identifies all jobs within the Enterprise Server.

Table 22-44 • AsJob Columns

Column	Type	Key	Nullable	Description
JobId	Int	Y	N	The primary key to uniquely identify a job.
JobName	Varchar(32)	N	N	The user provided name for the job.
UserName	Varchar(255)	N	N	The user name that created this job and derived as part of the Enterprise Server login process.
JobType	Varchar(5)	N	N	A string identifying the two job types of Custom (cust) and Template (temp) jobs.
Created	Datetime	N	N	The time when the job was created. Generated automatically.
Deleted_Flag	Char(1)	N	N	Not currently used.
Deleted	Datetime	N	Y	Not currently used.
Description	Varchar(400)	N	Y	An extended description used to elaborate the purpose of a (Template) job.

AsJobQueue

This table is the active queue for all job instances which are available to run, or which have run

Table 22-45 • AsJobQueue Columns

Column	Type	Key	Nullable	Description
JobQueueId	Int	Y	N	The primary key to uniquely identify a job queue entry.
QueueName	Varchar(32)	N	Y	The name of the job queue entry.

Table 22-45 • AsJobQueue Columns (cont.)

Column	Type	Key	Nullable	Description
LastAction	Varchar(200)	N	Y	The last status message generated by the job execution. Initially empty until the job actually runs.
Priority	Int	N	Y	Relative priority value giving preference to jobs scheduled for immediate execution
CompletionTime	DateTime	N	Y	The time that the last run of the job completed.
RequestTime	DateTime	N	N	The time after which the job will be set as available to run.
JobId_	Int	N	N	Foreign key into the AsJob table – JobId column. This value identifies which job is associated with this job queue entry.
UserName	Varchar(255)	N	N	The name of the user who added this job to the job queue and derived as part of the Enterprise Server login process.
MetaData	Varchar(4096)	N	N	Job specific XML data used by Template jobs.
StatusId_	Int	N	N	Foreign key into the AsQueueStatus table – StatusId column. This value identifies the status of this particular job queue entry.

AsJobStep

This table identifies the individual job steps for all jobs within the Enterprise Server.

Table 22-46 • AsJobStep Table Columns

Column	Type	Key	Nullable	Description
JobStepId	Int	Y	N	The primary key to uniquely identify a job.
JobStepName	Varchar(32)	N	N	The name of the job step.
UserName	Varchar(255)	N	N	The user name that created this job step and derived as part of the Enterprise Server login process.
MetaData	Varchar(4096)	N	N	Job step specific XML data

Table 22-46 • AsJobStep Table Columns (cont.)

Column	Type	Key	Nullable	Description
Created	Datetime	N	N	The time when the job step was created. Generated automatically.
Deleted_Flag	Char(1)	N	N	Not currently used.
Deleted	Datetime	N	Y	Not currently used.
Description	Varchar(400)	N	N	An extended description of the purpose of the job step.
JobId_	Int	N	N	Foreign key into the ASJob table indicating too which job this job step is associated.
JobOrder	Int	N	N	The ordinal number in which the job step is executed when the job itself is executed.
StepId_	Int	N	N	Foreign key into the AsJobStepMaster table which identifies the type of job.

Workflow Manager Tables

Information on Workflow Manager tables is provided in this section to assist you in writing queries to retrieve Workflow Manager data. This section includes a Workflow Manager database table schema diagram, as well as column information on each table.

Table 22-47 • Workflow Manager Tables

Table Name	Description
Workflow Manager Database Table Schema Diagram	Diagram that illustrates the relationship between the Workflow Manager tables.
AMS_Application Table	Stores information about Application Requests.
AMS_ApplicationItem Table	Stores information about each Data Group, Data Element, Workflow Phase and step that make an Application Request.
AMS_Person Table	Stores information about the Users who can access Workflow Manager.
AMS_ApplicationStatus Table	Stores information about the Application Request Statuses in Workflow Manager.
AMS_RelatedApplications Table	Stores information about related applications.

Table 22-47 • Workflow Manager Tables

Table Name	Description
AMS_Issues Table	Stores all the issues and their properties.
AMS_AuditTrail Table	Stores audit information of Application Requests.
AMS_Contracts Table	Stores Project related information.

Workflow Manager Database Table Schema Diagram

The following diagram illustrates the relationship between Workflow Manager tables:



Figure 22-21: Workflow Manager Database Table Schema Diagram

AMS_Application Table

This table stores the information about the Application Requests created in Workflow Manager. Some of the columns that can help you retrieve information are listed in the following table.

Table 22-48 • AMS_Application Table Column Information

Column	Description
ApplicationID	Primary Key of this table.
ApplicationLName	This column stores the name of the Application Request created by Customer or Servicer.
ContractID	Each Application is tied with a Project. This column stores the Project ID of the Project with which this application is tied with. More Information about this Project can be found from AMS_Contracts table using this Project ID.
UploadDate	This column stores the date and time when this Application Request was created.
UploadBy	This column stores the Person ID of the user who created this Application Request. More information about this user can be found from AMS_Person table using this Person ID.
DueDate	This column stores the Date & Time by which this application should be completed.
StatusSummary	<p>This column show the status of this Application Request. Following table show the status of the application based on the value in this column.</p> <ul style="list-style-type: none"> ● 0—Data Entry Phase ● 10—Workflow Phase ● 20—Acceptance Phase ● 30—Final Phase ● 90—Completed ● 40—AdminStudio Step ● 1010—Merge Workflow Step (used internally)
CurrentWFMajorItemID	The value in this column indicates which Workflow Phase is currently being executed for this Application Request. User can find further information about this Workflow Phase from ams_wfmajorItemtpl table.
CurrentWFMinorItemID	The value in this column indicates which Workflow Step is being executed for this Application Request. User can find further information about this Workflow Step from ams_wfminorItemTpl table.

Table 22-48 • AMS_Application Table Column Information (cont.)

Column	Description
TemplateID	This column stores the Template ID of the Template that used to create the Workflow of this Application Request. More Information about this Template can be found from AMS_Templates table using this Template ID.
ApplicationEndDate	This column stores the actual Date and Time when this Application Request was completed.
AppStatusID	This column is a foreign key from AMS_ApplicationStatus table and reflects the status of the Application. Status of the Application can be Active or Inactive by default and user can create additional statuses.

AMS_ApplicationItem Table

This table stores the information about each Data Group, Data Element, Workflow Phase and step that make an Application Request. Information like who completed a Workflow Step, when was it completed, is the data has been accepted by the Servicer, Total time spent on a step can be found in this table.

Table 22-49 • AMS_ApplicationItem Table Column Information

Column	Description
ApplicationItemID	Primary key of this table
ApplicationID	This value in this column indicates the Application that this Item belongs to. User can find further information about this application from AMS_Application table.
MajorMinorType	The value in this table indicates if the Item is a Workflow Phase (Major) or Workflow Step (Minor). Value of 1 in this column indicates Workflow Phase and value of 2 in this column indicates Workflow Step. In both the cases value in DataWFType will remain 2.
DataWFType	The value of 1 in this column indicates that this Application Item is either a Data Group or Data Element. If a value of 1 also exists in the MajorMinorType column, then the Application Item is a Data Group. If a value of 2 exists in the MajorMinorType column, then this Application Item is a Data Element.
CreateDate	This column indicates the Date and Time when this Application Item was created.
UpdatedBySubmit	This column contains the Person ID of the user who submitted a given Data Group or data Element.
LastUpdateSubmit	This column contains the Date & Time of when a Data Group or Data Element was submitted.

Table 22-49 • AMS_ApplicationItem Table Column Information (cont.)

Column	Description
UpdatedByAccept	This column contains the Person ID of the user who accepts the data submitted by the Customer.
LastUpdateAccept	This column contains the Date & Time of when the Data Submitted by the Customer was accepted by the Servicer.

AMS_Person Table

This table stores information about the Users who can access Workflow Manager. This table can store Workflow Manager database users and Directory Services Users/Groups.

Table 22-50 • AMS_Person Table Column Information

Column	Description
PersonID	Primary key of this table
CompanyID	This column stores the ID of the company where this user belongs. More information about the Company can be found from AMS_Company table.
FName	This column contains the First Name of the User
LName	This column contains the Last Name of the User
UserName	The value in this column is used to login into Workflow Manager. For LDAP users, this value is LDAP User Names (such as NT User name for Active Directory) and for Workflow Manager database Users the username is in email format.
Email	This column stores the email address of the user. This email address is used for sending notifications. If this column is empty then no emails are sent to this user.
Disabled	The value in this column indicates if the user is enabled or disabled. If the value is 1 then user is disabled. A user who is disabled cannot login into Workflow Manager.
DistinguishedName	This column is empty for non LDAP users. For users added through LDAP this is a unique value that can help identify the user in LDAP server quickly.

AMS_ApplicationStatus Table

By default, Workflow Manager comes with two default Application statuses: **Active** and **Inactive**. However, you can create additional statuses such as **Closed**, **Canceled**, and so on. The **AMS_ApplicationStatus** table stores the information about the Application Statuses in Workflow Manager. This status is then used by Application Requests in Workflow Manager.

Table 22-51 • AMS_ApplicationStatus Table Column Information

Column	Description
StatusID	Primary Key of this table
Status_Text	Name of the Status (such as Completed)
IsActive	If the value of this column is 1 then any application with this status is considered as Active and will be displayed in Workflow Manager UI. If the value of this column is 0 then Workflow Manager will not display this Application Request in Workflow Manager UI.

AMS_RelatedApplications Table

One Application can be related to one or more Applications. These applications might be duplicate or might accomplish the same goal. Workflow Manager stores the information of the related applications in AMS_RelatedApplications table.

Table 22-52 • AMS_RelatedApplications Table Column Information

Column	Information
ApplicationID	This column stores the ID of the Application that maybe related with one or more Applications.
Related_ApplicationID	This column stores the ID of Related Application. More information about Application can be found from AMS_Application table.

AMS_Issues Table

This table stores all the issues and their properties.

Table 22-53 • AMS_Issues Table Column Information

Column	Description
IssueID	This Column stores the primary key for the Issue Table and each issue can be uniquely identified by this number.
IssueAbstract	This stores the Inline description/Abstract of the issue.

Table 22-53 • AMS_Issues Table Column Information (cont.)

Column	Description
IssueText	This stores the detail information about the issue.
IssueType	<p>This is an integer value which stores the type of the issue. An issue can be of various types as follows:</p> <p>User Created Issues</p> <ul style="list-style-type: none"> • 1—Issue is only shared between Service Company users. • 2—General type of issue for Customers. • 3—General type of issue for Servicers. • 7—When user responds to any issue, a new issue is created of this issue type. • 8—Critical Issue: This will block the Workflow progress and requires special permission to create or close it. <p>System Generated Issues</p> <p>There are also some system generated issues. They are as follows:</p> <ul style="list-style-type: none"> • 4—When Update History step is completed in Workflow. • 5—When Data submitted is rejected by the Service Company user. • 6—When any Workflow Step is rolled back.
Status	<p>This column stores the status of the issue as integer values , they are as follows:</p> <ul style="list-style-type: none"> • 0—New Created Issue which are not read. • 1—Issue is already read. • 2—Issue is already responded. • 3—Issue is closed.
ApplicationID	This column stores the ID of the Application for which the issue is created. This ID can be used as join in AMS_Application table which will give information about the Application for the issue.
CreatedBy	This column stores the ID for the User who has created the issue. This ID is AMS_Person table primary key; hence this ID can be used as join in AMS_Person table to get more information about the user who created the issue.
CreateDate	This column stores the date and time (format: MM/DD/YYYY HH:MM:SS AM/PM) when the issue was created.
ReadBy	This column stores the ID for the User who has read the issue last (most current). This ID is AMS_Person table primary key; hence this ID can be used as join in AMS_Person table to get more information about the user who read the issue.

Table 22-53 • AMS_Issues Table Column Information (cont.)

Column	Description
ReadDate	This column stores the date and time (format: MM/DD/YYYY HH:MM:SS AM/PM) when the issue was last read.
ClosedBy	This column stores the ID for the User who has closed the issue. This ID is AMS_Person table primary key; hence this ID can be used as join in AMS_Person table to get more information about the user who closed the issue.
CloseDate	This column stores the date and time when the issue was closed, using the following format: MM/DD/YYYY HH:MM:SS AM/PM
OriginalIssueID	This column stores the IssueID (primary identifier for an issue) for the original issue of a responded issue.
WFMajorItemID	This column stores the ID of the Workflow Phase (major Workflow item) that the issue was created for. This ID is the AMS_WFMajorItemTpl table primary key; hence this ID can be used as join in AMS_WFMajorItemTpl table to get more information about this Workflow Phase.
WFMinorItemID	This column stores the ID of the Workflow Step that the issue was created for. This ID is the AMS_WFMinorItemTpl table primary key; hence this ID can be used as join in AMS_WFMinorItemTpl table to get more information about this Workflow Step.

AMS_AuditTrail Table

This table stores the audit information of an active Application Request and its data.

Table 22-54 • AMS_AuditTrail Table Column Information

Column	Description
AuditTrailID	Each new entry made in this table stores a unique GUID in this column. This is the primary key for the AMS_AuditTrail Table.
AuditTypeCode	This column contains values that indicate the events which generate an entry to the AuditTrail table. For a list of all of these values, see AuditTypeCode Column Values in the AMS_AuditTrail_Table
AuditTable	Stores the name of the Workflow Manager table that was audited.
AuditField	Stores the name of the column in the table that was audited.
AuditTablePK	Stores the primary key information of the table that was audited.

Table 22-54 • AMS_AuditTrail Table Column Information (cont.)

Column	Description
ApplicationID	This column stores the ID of the Application for which the Audit information was created. This ID can be used as join in AMS_Application table which will give information about the Application for the issue.
IssueID	This column stores the ID of the Issue for which the Audit information was created, if any related issue was also created for the same audit. This ID can be used as join in AMS_Issues table which will give information about the issue.
ChangedDate	This column stores the date and time (format: MM/DD/YYYY HH:MM:SS AM/PM) when the audit was created.
ChangedBy	This column stores the ID for the User whose actions created the audit. This ID is AMS_Person table primary key; hence this ID can be used as join in AMS_Person table to get more information about the user.
OldValue	This column stores the old data information of a data item of the active application Workflow, whose value is changed.
NewValue	This column stores the new data information of a data item of the active application Workflow, whose value is changed.
IterationCount	This gives the number of iteration an active application Workflow has been through. The value starts from 1 and increases with each rollback of the active application.
Comment	This column stores any comments for the audit (either system generated or entered by user).

AuditTypeCode Column Values in the AMS_AuditTrail_Table

This column contains the following values:

Table 22-55 • AuditTypeCode Values in the AMS_AuditTrail Table

Code	Description
100 / WFStep_Complete	When Workflow Step is completed for an active Application Request.
101 / WFStepRollback_User	When a Workflow Step is rolled back by a user.
102 / WFStepRollback_System	When a Workflow Step is rolled back by the system.
110 / WFStep_Cancel	When a Workflow Step is cancelled before completion.
201 / EmailIssue	When an email issue (general issue type) is created by user.
202 / EmailWFRollback	When email is sent to user for Workflow Step rollback.
203 / EmailDataUnlockOrReject	When email is sent to user for Data reject in Workflow Data Entry Step.
204 / EmailWFAdvance	When email is sent to user for Workflow Step advancement.
205 / EmailIssueRespond	When any issue is responded.
291 / DataSubmit_Begin	User opens data submission screen and begins entering data.
292 / DataAccept_Begin	User clicks Submit to submit requested data.
300 / DataChange	When data change is done in Data Entry step in Workflow.
301 / DataSubmit_Complete	When data submit is done any step in Workflow.
302 / DataAccept_Complete	When data acceptance is done in Data Entry step in Workflow.
303 / DataUnlock	When data unlock is done in data entry step in Workflow. (This error code only used in versions previous to AMS 2.5.)
304 / DataReject	When data rejection is done in Data Entry step in Workflow.
305 / AppNameChange	When Application Request name change is done.
306 / DataEdit	When an active Application Request Workflow Step is edited.
312 / DataAccept_Cancel	When a Data Acceptance step is cancelled before completion.
400 / SLA_Start	Indicates the start of an Application Request's recorded SLA period.

Table 22-55 • AuditTypeCode Values in the AMS_AuditTrail Table (cont.)

Code	Description
401 / SLA_Stop	Marks the completion of an Application Request, indicating the end of its recorded SLA period. Using codes 401 and 402, you can determine an Application Request's elapsed SLA time.
500 / WFPhase_Begin	When a Workflow Phase is started.
501 / WFPhase_Complete	When the last step of a Workflow Phase has been completed.
502 / WFPhase_Cancel	When a user cancels a Workflow Phase.
600 / UserManagement	When user management activity occurs.
700 / WorkAssignment	When a work assignment is made.

AMS_Contracts Table

This table stores the audit information of the live Application and its data.

Table 22-56 • AMS_Contracts Table Column Information

Column	Description
ContractID	This column stores the primary key value for this table which can identify each contract (project) uniquely.
ContractName	This column stores the name of the contract (project).
StartDate	This Date is start day for the project.
EndDate	This column stores the end date for the project/contract.
Disabled	This column stores the status of the project: 0 —Active 1 —Inactive.
IncludeWeekends	This column stores the default value for all the Application created from this project , whether weekends will be included in calculation of due date.
DefaultDuePeriod	This column stores the default due period value for all the Application created from this project.
DefaultRiskPeriod	This column stores the default Risk period value for all the Application created from this project.

Table 22-56 • AMS_Contracts Table Column Information (cont.)

Column	Description
RequestPrefix	This column stores the Request Prefix value for all the Application created from this project.
CompanyID	CompanyID column stores the ID of the company for the project. More information about the Company can be found from AMS_Company table.
TemplateID	This column stores the Template ID of the Template that used to create the Workflow for the Application Request created from this project. More Information about this Template can be found from AMS_Templates table using this Template ID.
DeploymentTemplateID	This column stores the Template ID of the Deployment Template that used to create the Deployment Workflow for the Application Request created from this project. More Information about this Template can be found from AMS_Templates table using this Template ID.

Automating Tasks Using Job Manager



Edition • Job Manager is included with AdminStudio Enterprise Edition.

You can use Job Manager to automate time consuming tasks such as package import, validation, conflict analysis, and conflict resolution. With Job Manager, you can schedule these tasks to run automatically at scheduled times, perhaps when system availability is highest. This enables your organization to increase the efficiency of your packaging process and to enforce standardized business practices.

Table 23-1 • Job Manager User Documentation

Section	Description
About Job Manager	Describes the purpose of Job Manager, and the types of tasks that you can automate.
Directory Monitoring and Job Manager	Describes how the Directory Monitoring Job Step can be included in both Template Jobs and Custom Jobs to examine a specified directory for new or modified packages and then to automatically import or reimport those packages into the Application Catalog.
Managing Jobs	Explains how to create a new job, reschedule a job, set email alerts for a job, and view job status.
Managing Templates	Explains how to create and edit Job Templates.
Setting Job Manager Conflict Detection Options	Explains how to set import, validation, and detection options.
Managing Jobs Using Job Manager Engine	Explains how to use Job Manager Engine to manage jobs from multiple Application Catalogs.

Table 23-1 • Job Manager User Documentation (cont.)

Section	Description
Job Manager Reference	Provides detailed information on all of the Job Manager pages.

About Job Manager

You can use Job Manager to automate time consuming tasks such as package import, validation, conflict analysis, and conflict resolution. Job Manager provides:

- **Increased efficiency**—With Job Manager, you can schedule common tasks—such as package import, conflict analysis, and conflict resolution—to run automatically at scheduled times (such as when system availability is highest), and no human intervention is required.
- **Enforcement of consistent business practices**—Job Manager enables organizations to enforce consistent business practices in the deployment preparation process. It allows you to perform the exact same series of steps on any package before it is deployed in your enterprise, such as checking against specific ICE or ACE rules, and against a specific operating system.
- **Dynamic group content identification**—You can create a Job that will perform actions on whatever packages are in a selected group at the time the Job is executed. You can schedule a reoccurring Job to automatically perform actions on the exact contents of a group without ever having to select any packages or create any new Jobs. This allows you to enforce consistent business practices on a group's contents even as the content changes.

This section includes the following topics:

- [Job Steps: Tasks You Can Automate](#)
- [Template Jobs vs. Custom Jobs](#)
- [Selecting Contextual Packages in a Custom Job](#)
- [Dynamic Group Content Identification](#)
- [Conflict Resolution and the Software Repository](#)
- [User Permissions and Job Manager](#)

Job Steps: Tasks You Can Automate

Jobs are comprised of one or more Job Steps. The following Job Step Types are available:

Table 23-2 • Job Manager Step Types

Task	Description
Scan for Dependencies	The Scan for Dependencies Job Step analyzes a Windows Installer package for dependent files. This information is used primarily to enhance the quality of any future Patch Impact Analysis operation that may be performed on this package.
Directory Monitoring	You can use the Directory Monitoring Job Step to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files, and if changes are detected, those new or modified files are then either imported or reimported into the Application Catalog. See About Automatically Importing Packages from a Network Directory for more information.
Validation	The Validation Job Step verifies the internal data integrity of a Windows installer package.
Conflict Detection	The Conflict Detection Job Step performs conflict detection against selected packages.
Import	The Import Job Step imports a specified package (of any package type except Merge Modules) into the Application Catalog. An Import Job Step can only be included in a Custom Job, not a Job based on a Template.
Best Practice Detection	The Best Practice ACEs Job Step operates against a single package to enforce Microsoft Windows Installer standards.
Resolution	The Resolution Job Step performs automatic conflict resolution on one Windows Installer package or a set of packages. To configure the Resolution Job Step, you first select the packages to perform conflict resolution on, and then you select at least one resolution.

About Automatically Importing Packages from a Network Directory

In AdminStudio Application Manager, you can use the **Network Directory** option of **Package Auto Import** to monitor a directory location on a network (or a local directory) and automatically import or re-import packages in that directory at scheduled intervals.

You can use this **Network Directory** option in combination with the **Directory Monitoring** Job Step to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files. If changes are detected, those new or modified files can then be either imported or reimported into the Application Catalog.



Note • For more information, see [Automatically Importing Packages from a Network Directory](#).

Template Jobs vs. Custom Jobs

When you create a new Job, you can either base it on an existing Template that has pre-defined Job Steps, or build a Custom Job by choosing which Job Steps you want to perform and in what order. Each type of Job serves a particular purpose and provides certain benefits.

Template Jobs

Creating Jobs based on pre-defined Templates enables an organization to enforce consistent business practices. System Administrators can create a Template to define a series of Job Steps that they always want performed on a certain type of package before it is deployed. Then, when a package that falls into that category needs to be prepared for deployment, a Job based on that Template can be created and run.

For example, if all users of a specific department of an organization use the same disk image (Operating System and set of applications), all packages that are distributed in that department would need to have conflict analysis performed against that OS Snapshot and those target applications. The System Administrator could create a Template that uses those specific target applications, and could include very specific ICE and ACE rules in the conflict analysis, based upon requirements specific to that target environment.

In Template Jobs:

- **Target packages** are selected when a Template is created.
- **Source packages** are selected when a Job based on a Template is created.
- **An Import Job Step cannot be included.**
- **A Directory Monitoring Job Step can be included** to examine the specified directory for new or modified packages and then automatically import or reimport those packages into the Application Catalog. This is equivalent to using the Application Manager **Package Auto Import Wizard** to automatically import packages from a network directory. However, subsequent Job Steps cannot perform any actions on packages imported or reimported as part of a Directory Monitoring Job Step.



Note • See [About Automatically Importing Packages from a Network Directory](#) for more information.

Custom Jobs

When creating a new Custom Job, you can include only the Job Steps that you want to perform, in the order you want to perform them, using the source and target packages that you select when you create the Job. In Custom Jobs:

- **Both Source and Target packages are selected** when the Custom Job is created.
- **An Import or Directory Monitoring Job Step can be included**, and you can use packages imported as part of those steps as Source packages in subsequent Job Steps.

Because you can insert an **Import** or **Directory Monitoring** Job Step, a Custom Job is well-suited to automate a bulk import process.

Selecting Contextual Packages in a Custom Job

When defining a Custom Job, the packages that you import in prior **Import** or **Directory Monitoring** Job Steps are available for selection as source or target packages when defining operations in subsequent Job Steps.

The packages that will be imported in an **Import** or **Directory Monitoring** Job Step are selected when the Job is executed, not when the Job Step is defined, so the names of those packages are unknown.

Therefore to make these imported packages available for selection when defining subsequent Job Steps, they are listed in the **Contextual Packages** group in the package tree and are referred to in context of their associated import step, such as Job Step n , where n is the number of the Job Step that performed the import.

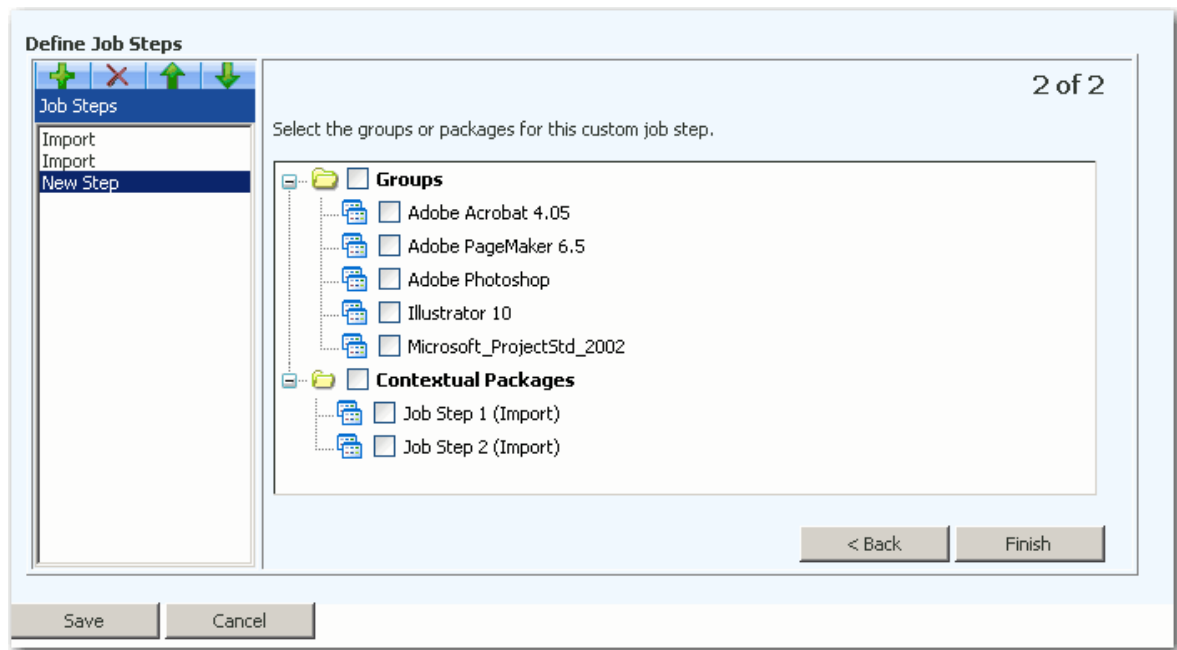


Figure 23-1: Contextual Packages Group

Dynamic Group Content Identification

You can create a Custom or Template Job that will perform actions on whatever packages are in a selected group *at the time the Job is executed*.

Therefore, you can schedule a reoccurring Job to automatically perform actions on the exact contents of a group without ever having to select any packages or create any new Jobs. This allows you to enforce consistent business practices on a group's contents even as the content changes.

For example, your organization could have a policy of importing all new packages into a specific group, such as a New Products Group. You could then choose to schedule a Job to run each night that would automatically perform conflict analysis and resolution using the packages in the New Products Group as the source packages and using selected packages in the Application Catalog as the target packages. In the morning, you could quickly see the current state of each package and continue the evaluation process.


Conflict Resolution and the Software Repository

If you include a **Conflict Detection** Job Step in a Job to identify conflicts, you can also include a **Resolution** Job Step to automatically resolve the conflicts that have associated Conflict Application Resolution Definitions (CARs), and then reimport the modified package into the Application Catalog.

Resolution Behaviors

Because conflict resolution requires that the package be modified, if a package is stored in the Software Repository and is currently checked-out by another user, it cannot be modified, and therefore cannot be resolved. This resolution behavior is detailed in the following table.

Table 23-3 • Job Manager Resolution Behavior Regarding the Software Repository

Package State	Resolution Behavior
Package in Software Repository / Checked Out	Job Manager ignores the resolution process for these packages. The packages are already being edited by someone else.
Package in Software Repository / Not Checked Out	<p>The package will be checked out by the virtual user Job Manager Engine prior to the normal operation of the resolution process.</p> <div> Checked Out Job Manager Engine Friday, August 05, 2005 - 11:35 AM</div> <p>If the resolution process results in a no-operation or the user does not elect to re-import the package, then the check-out operation will be cancelled.</p>
Package Not in Software Repository	The resolution process will operate normally.

Re-Import Behaviors

When defining a Resolution Job Step, you have to indicate the reimport option for resolved packages. When defining a **Resolution** Job Step, you have the choice of three re-import options for resolved packages: re-import as a new version, re-import by overwriting the existing version, or do not re-import.

The choice you make depends upon whether the package is stored in the Software Repository:

Table 23-4 • Resolved Package Re-Import Options

Resolved Packages Re-import options	Package in Software Repository	Package Not in Software Repository
As a new version	Package will be re-imported and checked-in as a new version.	N/A Package cannot be imported as a new version unless it is in the Software Repository.
By overwriting existing version	Package will be re-imported and checked-in, replacing the existing version.	Package will be re-imported, replacing the original package.
Do not import the resolved packages	Package will not be re-imported.	Package will not be re-imported.

User Permissions and Job Manager

The list of Job Step types that are available for you to select when creating a Job is controlled by the permissions of the Roles you are assigned to. For example, if you are not assigned to a Role that includes conflict resolution permission, the **Resolution** Job Step will not be available for your selection.

You can obtain access to the complete list of Job Step types by being assigned to additional Roles or having the Roles that you are assigned to modified to provide additional permissions.

Directory Monitoring and Job Manager

A **Directory Monitoring** Job Step can be included in both Template Jobs and Custom Jobs to examine a specified directory for new or modified packages and then to automatically import or reimport those packages into the Application Catalog. This is equivalent to using the Application Manager **Package Auto Import Wizard** to automatically import packages from a network directory.



Note • See [About Automatically Importing Packages from a Network Directory](#) for more information.

However, the use of these imported packages varies between Template Jobs and Custom Jobs:

- **Template Jobs** can include a **Directory Monitoring** Job Step to import packages, but subsequent Job Steps in that Job *cannot* perform any actions on any packages that are imported/reimported as part of that Directory Monitoring Job Step.

- **Custom Jobs** can include a **Directory Monitoring** Job Step, and you *can* use packages imported as part of that Directory Monitoring Job Step as source packages in subsequent Job Steps. See [Selecting Contextual Packages in a Custom Job](#).

Managing Jobs

This section explains how to create and submit Jobs for execution. Topics include how to create and edit Template and Custom Jobs, reschedule a Job, set email alerts for a Job, and view Job status. The following topics are included:

- [Creating a New Job](#)
- [Editing an Existing Job](#)
- [Rescheduling a Job](#)
- [Setting Email Alerts](#)
- [Viewing Job Status](#)
- [Deleting a Job](#)

Creating a New Job

When creating a new Job to execute, you have the choice of either creating a Job based on a Template or a new Custom Job:

- **Template Job**—A Job based on a Template consists of a series of pre-defined Job Steps. See [Creating a New Job Based on a Template](#).
- **Custom Job**—Allows you to include only the Job Steps that you want to perform, in the order you want to perform them, and using the source and target packages that you select. Custom Jobs can include the **Import** Job Step, while Template Jobs cannot. See [Creating a New Custom Job](#).

Creating a New Job Based on a Template

A Template Job consists of a series of pre-defined Job Steps that are performed on the package(s) or group(s) you select from the Application Catalog.

To create a job based on a Template, perform the following steps.



Task: *To create a Job based on a Template:*

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.
3. Click the **Add** button. The **Add Job** page opens with the **Template** option selected.

4. From the **Template** list, select the appropriate Template.
5. In the **Name** field, enter a name for this job.
6. In the **Package List** tree, select the package(s) or group(s) that this Job will perform actions on. For example, if you chose a Job Template named *Scan_Dependencies_Template*, select the packages that you want to scan.

If you choose a group, this Job's actions will be performed on all the packages in that group *at the time the Job is run*. With this **dynamic group content identification** feature, you can schedule a reoccurring Job to automatically perform actions on the exact contents of a group without ever having to select any packages or create any new Jobs.

7. **Schedule** this Job by selecting one of the following options from the Execution list:

- **Immediate**—Execute this Job immediately.
- **Scheduled**—Execute this Job at the **Date** and **Time** selected.



Note • You can also choose to set this job to reoccur at a specified interval. See [Rescheduling a Job](#)

8. Click **Save**. The new Job is now listed on the **Jobs Queue** page and will be run at its scheduled time.

Creating a New Custom Job

When you create a Custom Job, you can include only the Job Steps that you want to perform, in the order you want to perform them, and using the source and target packages that you select. Custom Jobs can include the **Import** Job Step, while Template Jobs cannot.

To create a Custom Job, perform the following steps.



Task: **To create a new Custom Job:**

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.
3. Click the **Add** button. The **Add Job** page opens with the **Template** option selected.
4. Enter a **Name** for this new job.
5. Clear the **Template** option. An empty **Define Job Steps** area opens.

Add Template

Name:

New Template

Description:

Define Job Steps

Job Steps

New Step

The Step Wizard will help you create, view, or modify the selected job step. To start, select the job step type below and click Next to continue.

Step Type:

Scan for Dependencies

Step Type Description:

The Scan for dependencies Job step will analyze a MSI package for dependent files. This information is used primarily to enhance the quality of any future Patch Impact Analysis operation performed on this package.

'Scan for Dependencies', 'Validation', and 'Best Practices Detection' template steps require no additional data; just click the 'Next' button below to complete the template step.

Next >

Save

Cancel

6. Select one of the listed tasks from the **Step Type** list:

- **Scan for Dependencies**—Analyze a Windows Installer package for dependent files.
- **Directory Monitoring**—Use to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files, and if any changes are detected, those new or modified files are then either imported or re-imported into the Application Catalog.
- **Validation**—Verifies the internal data integrity of a Windows Installer package.
- **Conflict Detection**—Performs conflict detection on selected packages using Conflict ACE Rules
- **Import**—Import a selected package into the Application Catalog.
- **Best Practice Detection**—Performs conflict detection on a single package using Best Practice ACE Rules to enforce Windows Installer standards.
- **Resolution**—Performs automatic conflict resolution on selected packages.

Each time you select a **Step Type**, you are prompted to specify additional information:

Step Type	Required Selections
Scan for Dependencies	<div><div>1. Click Next. The Step 2 of 2 View opens.</div><div>2. Select the packages or groups of packages that you want to scan, and then click Finish to save this Job Step.</div></div>

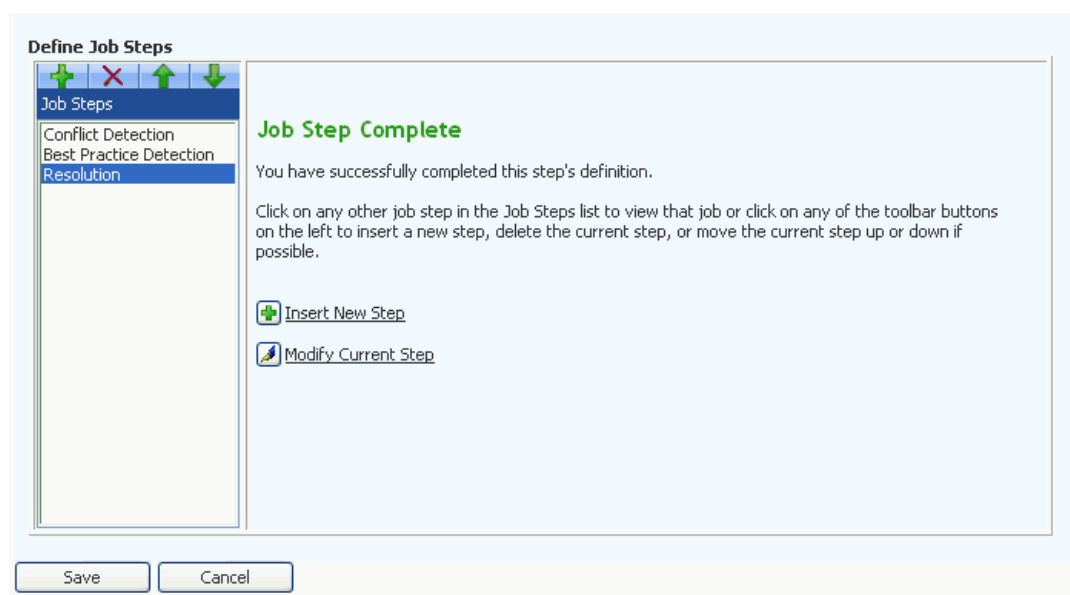
1762

ADS-0950-UG01 AdminStudio 9.5 User Guide

Step Type	Required Selections
Directory Monitoring	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. In the Directory Name box, enter the UNC path of the directory that you want to monitor. 3. Choose whether to Include subdirectories in the 'Directory Monitoring' process. 4. Click Finish to save this Job Step.
Validation	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. Select the packages or groups of packages that you want to validate. 3. Click Finish to save this Job Step.
Conflict Detection	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 4 view opens. 2. Select the source packages or groups of packages that you want to include in the conflict detection process, and then click Next. The Step 3 of 4 View opens. 3. Select the target packages or groups of packages that you want to use in the conflict detection process and click Next. The Step 4 of 4 View opens. 4. Select the set of Conflict ACE Rules that you want to include in the conflict detection process and click Finish to save this Job Step.
Import	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 4 View opens. 2. Select a package to import. 3. Specify whether to include the package in the Software Repository 4. Click Next. The Step 3 of 4 View opens. 5. Select any transform (.mst) or patch (.msp) files to import. 6. Click Next. The Step 4 of 4 View opens. 7. Select one or more destination groups into which the package will be imported. 8. Click Finish to save this Job Step.
Best Practice Detection	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. Select the packages or groups of packages that you want to perform Best Practice conflict detection on. 3. Click Finish to save this Job Step.

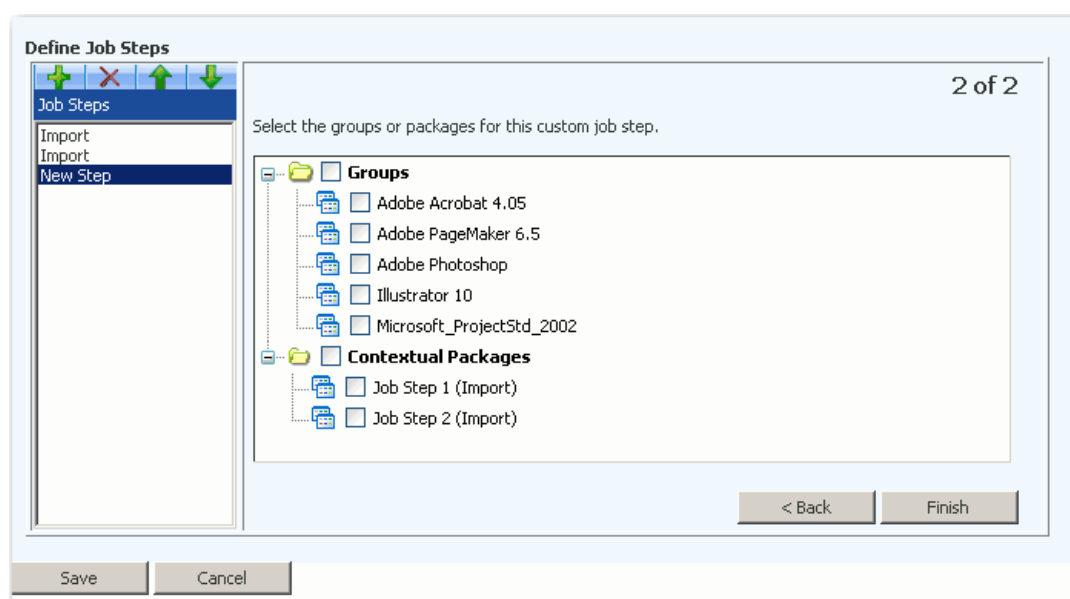
Step Type	Required Selections
Resolution	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 3 View opens. 2. Select the packages or groups of packages that you want to perform conflict resolution on, and then click Next. The Step 3 of 3 View opens. 3. Select the Conflict ACE Rules that you want to resolve automatically. 4. Select a Resolved packages re-import option. See Re-Import Behaviors. 5. Click Finish to save this Job Step.

7. Each time you click **Finish**, the new Job Step is saved and you are prompted to add another Job Step.



8. Continue adding desired Job Steps by clicking **Insert New Step**.
9. If your Job includes **Import** or **Directory Monitoring** Job Steps, you can perform actions on those imported or reimported packages in subsequent Job Steps. For example, if you add a **Validation** Job Step after an **Import** or **Directory Monitoring** Job Step, you can choose to validate the newly imported/reimported package or packages from the previous Steps.

To make these packages available for selection when defining subsequent Job Steps, they are listed in the **Contextual Packages** group in the package tree and are referred to in context of their associated step, such as Job Step n , where n is the number of the Job Step that performed the import.



Note • During a **Directory Monitoring** Job Step, one or more packages in the monitored directory may be imported or reimported into the Application Catalog. Therefore, if you select a **Directory Monitoring** Job Step as a Source for a subsequent Job Step, that Step's action will be performed on all of the newly imported or reimported packages.

10. **Schedule** this Job by selecting one of the following options from the Execution list:

- **Immediate**—Execute this Job immediately.
- **Scheduled**—Execute this Job at the **Date** and **Time** selected.



Note • You can also choose to set this job to reoccur at a specified interval. See [Rescheduling a Job](#)

11. Click **Save**. The new Job is now listed on the **Jobs Queue** page and will be run at its scheduled time.

Editing an Existing Job

You can edit an existing Job by selecting the Job on the **Jobs Queue** page to open the **Edit Job** page. The following information can be edited:

Table 23-5 • Editable Items in an Existing Job

Job Step Type	Editable Items
General	<ul style="list-style-type: none">• Job Name• Execution options• Step Type
Scan for Dependencies Job Step	<ul style="list-style-type: none">• Selected packages and groups
Directory Monitoring Job Step	<ul style="list-style-type: none">• Directory to monitor• Option to monitor subdirectories
Validation Job Step	<ul style="list-style-type: none">• Selected packages and groups
Conflict Detection Job Step	<ul style="list-style-type: none">• Selected source packages and groups• Selected target packages and groups• Set of conflict detection rules to use during the conflict detection process
Import Job Step	<ul style="list-style-type: none">• Package to import• Software Repository option• Transform (.mst) and Patch (.msp) files to include during import• Destination group for imported package
Best Practice Detection Job Step	<ul style="list-style-type: none">• Selected packages and groups
Resolution Job Step	<ul style="list-style-type: none">• Selected packages and groups• Conflicts to resolve• Resolved package re-import options

To edit an existing Job, perform the following steps:



Task: *To edit an existing Job:*

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.

3. Select the Job that you want to edit. The **Edit Job** page opens.
 - Only one Job can be edited at a time.
 - You cannot edit a Job with a **Status** of **Running** or **Pending**.
 - You cannot change a **Custom Job** to a **Template Job** or vice versa.
4. Edit the Job following the instructions in [Creating a New Job Based on a Template](#) or [Creating a New Custom Job](#).
5. When you have finished editing the Job, click **Save**.

Rescheduling a Job

You can reschedule a completed Job so that it is run again. You can also reschedule Jobs that failed, were cancelled, or are scheduled to be run in the future.

If a Job is in **Scheduled**, **Complete**, **Failed**, or **Cancelled** Status, open the **Jobs Queue** page and click the link in the **Schedule** column for that Job. The **Schedule Job** page opens.



Task:

To reschedule a job:

1. On the **Jobs Queue** page, click the date and time listed in the **Schedule** column for the Job you want to reschedule. The **Schedule Job** page opens.
2. Select either **For Immediate Execution** or **For Scheduled Execution**.
 - If you selected **For Immediate Execution**, click **Save** and your Job will be executed immediately.
 - If you selected **For Scheduled Execution**, continue with step 4.
3. Click the calendar icon next to the **Select Date** field and select the date when you want the Job to be run.
4. In the **Enter Time** fields, select an hour of the day from the first list, select the minutes after the hour from the second list, and select **AM** or **PM** from the third list.
5. If you want this Job to reoccur at a specified interval, set the **Recurrence Interval** option to **Daily**, **Weekly**, or **Monthly**, and perform the following steps:
 - a. If you want this Job to reoccur a specific number of times, select the **Number of Times** option, and enter a number in the box.
 - b. If you want this Job to reoccur until a specified expiration date, select the **Expiration Date** option, click the calendar icon, and select a date from the calendar.
6. Click **Save**.

Setting Email Alerts

You can choose to send an email Alert when a Job is completed and when it fails. To do this, open the **Jobs Queue** page, and click **Add** in the **Alerts** column for the Job that you want to set an alert for. You then can create an email message on the **Job Alerts** page.



Tip • You can choose to send a Job alert upon both Job failure and Job success.

When composing the **Subject** line and the **Message**, you can also insert [System Variables](#) to instruct Job Manager to automatically insert the variable data regarding that Job at runtime.



Task: To set email alerts:

1. On the **Jobs Queue** page, click **Add** in the **Alerts** column for the Job that you want to set an alert for. The **Job Alerts** page opens.
2. From the **Send the following e-mail alert when the job** list, select **fails** or **succeeds**.
3. In the **To** field, enter the email addresses that you want to be notified when a Job is completed or when it fails. Separate multiple email addresses with a semicolon (;).
4. In the **Subject** box, enter text to describe the purpose of the email. You can select [System Variables](#) from the list so that the value of the variable for this Job is inserted when the email is sent.
5. In the **Message** box, enter the text of the message. You can select [System Variables](#) from the list so that the value of the variable for this Job is inserted when the email is sent.
6. Click **Save**.

System Variables

The following system variables are available to use in email alerts:

Table 23-6 • System Variables Available to Use in Email Alerts

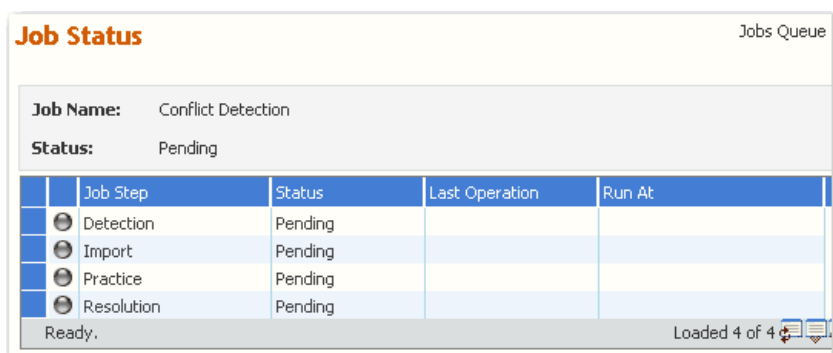
Variable	Description
QueueName	Name of Job.
LastAction	The last Job Step that was completed for this Job. If a Job's status is Running, this is the Job Step that is currently being executed. If a Job's status is Complete, this is the last Job Step of the Job. If a Job Failed, this is the Job Step that was running when the failure occurred.
CompletionTime	Time job was completed.
UserName	Person who submitted the Job.

Table 23-6 • System Variables Available to Use in Email Alerts

Variable	Description
JobName	Name of Template.
JobType	Number of Job Steps defined in the Template.
Description	Description that was entered to identify the Template.
ProductVersion	MSI specific property from package.
ProductName	MSI specific property from package.
Manufacturer	MSI specific property from package.
ProductLanguage	MSI specific property from package.
FileName	Base file path (either “Managed from the Software Repository” or appropriate path).
Comments	Product comments.
ImportDate	Date package was imported into the Application Catalog.
DisplayedProductName	Product name as displayed in Application Manager/ConflictSolver.
OriginalMsiFileName	Name of .msi file that was imported into the Application Catalog.
OriginalPackageLocation	Location of the original .msi file at the time it was imported into the Application Catalog.

Viewing Job Status

On the **Jobs Queue** page, a Job’s current status is listed as a hyperlink in the **Status** column. If you click on this hyperlink, the **Job Status** page opens, displaying the Job Name, the Job Status, and the status of each of the Job Steps in that job.



Job Step	Status	Last Operation	Run At
Detection	Pending		
Import	Pending		
Practice	Pending		
Resolution	Pending		

The possible statuses are:

- **Scheduled**—Job is scheduled to be executed at a future date and time.
- **Running**—Job is currently being run. Only one job per Application Catalog connection will ever be running at one time.
- **Pending**—Job is available to run but is waiting for the running job to complete. A job set for immediate execution will be automatically assigned a Pending job status.
- **Complete**—Job has been executed.
- **Canceled**—Job was canceled before it was executed or before its execution was complete.
- **Failed**—Job was unable to run either because of a software error, a network communication problem, a hardware failure, or any other reason.

Deleting a Job

You can delete a Job by opening the **Jobs Queue** page and clicking **Delete** in the **Action** column of the Job you want to delete.



Note • Deleting a Template Job in no way effects the Job Template that it was based on.



Task:

To delete a Job:

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.
3. Click **Delete** in the **Action** column of the Job that you want to delete. The **Delete Job** page opens and you are prompted to confirm the deletion.



Note • You cannot delete a Job with a **Status** of **Running**.

4. Click **Delete**. The Job is no longer listed on the **Jobs Queue** page.

Managing Templates

Creating Jobs based on pre-defined Templates enables an organization to enforce consistent business practices. System Administrators can create a Template to define a series of Job Steps that they always want performed on a certain type of package before it is deployed. Then, when a package that falls into that category needs to be prepared for deployment, a Job based on that Template can be created and run.

For example, if all users of a specific department of an organization use the same disk image (Operating System and set of applications), all packages that are distributed in that department would need to have conflict analysis performed against that OS Snapshot and those target applications. The System Administrator could create a Template that uses those specific target applications, and could include very specific ICE and ACE rules in the conflict analysis, based upon requirements specific to that target environment.

This section includes the following topics:

- [Source and Target Packages in Template Jobs vs. Custom Jobs](#)
- [Creating a New Template](#)
- [Editing an Existing Template](#)
- [Deleting a Template](#)

Source and Target Packages in Template Jobs vs. Custom Jobs

The selection of source and target packages in a Template Job differs from how source and target packages are selected in Custom Jobs.

For Template Jobs:

- **Target packages** are selected when a Template is created.
- **Source packages** are selected when a Job based on a Template is created.
- **An Import Job Step cannot be included.** All actions must be performed on packages that have already been imported into the Application Catalog.

For Custom Jobs:

- **Both Source and Target packages** are selected when the Custom Job is created.
- **An Import Job Step can be included**, and you can use newly imported packages as Source packages in subsequent Job Steps.

Creating a New Template

You create a Template much like the steps described in [Creating a New Custom Job](#), except:

- You do not **Schedule** a Template
- You do not set **Alerts** for a Template
- You cannot include an **Import** Job Step in a Template.

You create Templates on the **Manage Templates** tab. The **Job Template List** page lists all defined Templates. From this page, you can **Add** a new Template, **Edit** an existing Template, or **Delete** a Template.



Task: *To create a new Template:*

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.
3. Click the **Manage Templates** subtab. The **Job Template List** page opens.
4. Click the **Add** button. The **Add Template** page opens, displaying an empty **Define Job Steps** area.

Add Template

Name:

Description:

Define Job Steps

1 of 1+

The Step Wizard will help you create, view, or modify the selected job step. To start, select the job step type below and click Next to continue.

Step Type:

Step Type Description: The Scan for dependencies Job step will analyze a MSI package for dependent files. This information is used primarily to enhance the quality of any future Patch Impact Analysis operation performed on this package.

'Scan for Dependencies', 'Validation', and 'Best Practices Detection' template steps require no additional data; just click the 'Next' button below to complete the template step.

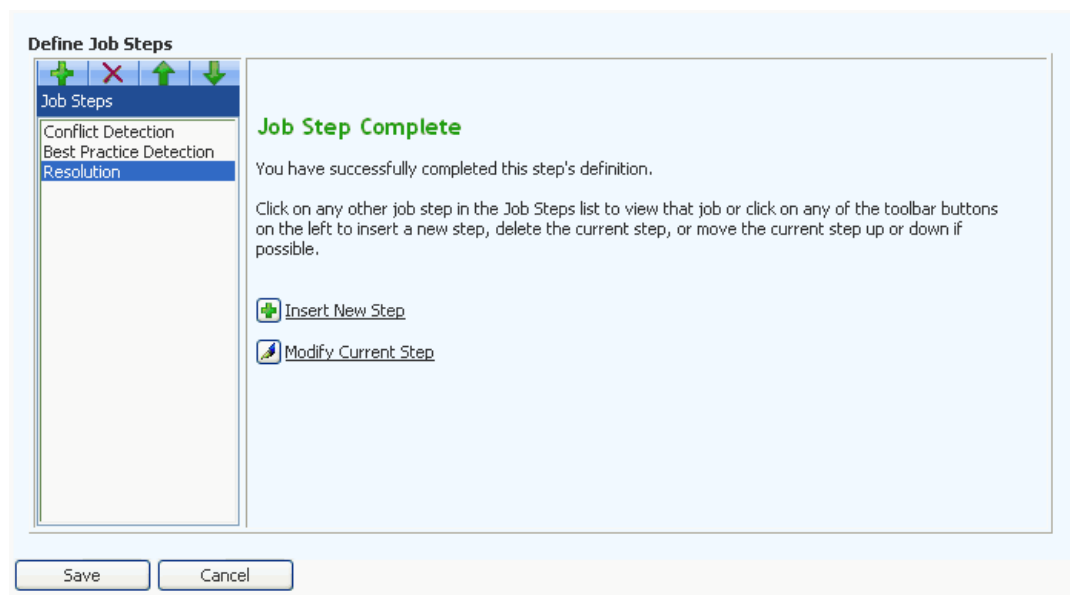
5. Enter a **Name** for this new Template.
6. Enter a **Description** to define the purpose of this new Template.
7. Select one of the listed tasks from the **Step Type** list:

- **Scan for Dependencies**—Analyze a Windows Installer package for dependent files.
- **Directory Monitoring**—Use to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files, and if any changes are detected, those new or modified files are then either imported or re-imported into the Application Catalog.
- **Validation**—Verifies the internal data integrity of a Windows Installer package.
- **Conflict Detection**—Performs conflict detection on selected packages using Conflict ACE Rules
- **Best Practice Detection**—Performs conflict detection on a single package using Best Practice ACE Rules to enforce Windows Installer standards.
- **Resolution**—Performs automatic conflict resolution on selected packages.

After you select a **Step Type**, you may be prompted to specify additional information. Make the required selections, as listed in the following table, and click **Finish** to save the Step.

Step Type	Required Selections
Scan for Dependencies	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)
Directory Monitoring	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 view opens. 2. In the Directory Name box, enter the UNC path of the directory that you want to monitor. 3. Choose whether to Include subdirectories in the 'Directory Monitoring' process. 4. Click Finish to save this Job Step.
Validation	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)
Conflict Detection	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 3 view opens. 2. Select the target packages or groups of packages that you want to use in the conflict detection process and click Next. The Step 3 of 3 view opens. 3. Select the set of Conflict ACE Rules that you want to include in the conflict detection process and click Finish to save this Job Step.
Best Practice Detection	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)
Resolution	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 view opens. 2. Select the Conflict ACE Rules that you want to resolve automatically. 3. Select a Resolved packages re-import option. See Re-Import Behaviors. 4. Click Finish to save this Job Step.

- Each time you click **Finish**, the new Job Step is saved and you are prompted to add another Job Step.



- Continue adding desired Job Steps.
- When you have finished adding Job Steps, click **Save**. The Template is now listed on the **Job Template List** page and in the **Template** list on the **Add Job** page.

Editing an Existing Template

You can edit an existing Template by selecting the Template on the **Job Template List** page to open the **Edit Template** page.



Task: *To edit an existing Template:*

- Login to AdminStudio Enterprise Server.
- Click the **Job Manager** tab. The **Jobs Queue** page opens.
- Click the **Manage Templates** tab. The
- Select the Template you want to edit. The **Edit Template** page opens.



Note • Only one Template can be edited at a time.

- Edit the Template following the instructions in [Creating a New Template](#).
- When you have finished editing the Template, click **Save**.

Deleting a Template

You can delete a Template by opening the **Job Template List** page and clicking **Delete** in the **Action** column of the Template you want to delete.

Only Templates which are not in use are allowed to be deleted.

- A Template is considered in use as long as a Template Job associated with the Template exists.
- If a Template Job associated with this Template is scheduled to be run in the future or if it is already completed, you will not be permitted to delete the Template.



Task: *To delete a Template:*

1. Login to AdminStudio Enterprise Server.
2. Click the **Job Manager** tab. The **Jobs Queue** page opens.
3. Click the **Manage Templates** tab. The **Job Template List** page opens.
4. Click **Delete** in the **Action** column of the Template that you want to delete. The **Delete Template** page opens, prompting you to confirm the deletion.
5. Click **Delete**. The Template is no longer listed on the **Job Template List** page.

Setting Job Manager Conflict Detection Options

You select the rules that you want Job Manager to use by default during conflict detection on the **Conflict Detection Options** page of the **Configuration tab**.



Note • By setting these Conflict Detection options, you are only setting the defaults for the Conflict Detection Job Step in a Job. When you are defining the Conflict Detection Job Step, you can change these selections.



Task: *To set Detection options:*

1. Login to AdminStudio Enterprise Server.
2. Click the Job Manager tab. The **Jobs Queue** page opens.
3. Click the **Configuration** subtab. The **Conflict Detection Options page** opens.

4. In the **Conflict Types** tree, select the check box for ACE rules that you want to be performed during conflict identification. Conflict rules (ACEs) associated with unselected boxes will not be performed during conflict identification. The following rules are available:

Category	ACEs
Components	<ul style="list-style-type: none"> • ACE30—A check for KeyPath conflicts across components. • ACE02—Check destination directories for identical components. • ACE09—Check for different merge module versions.
File Extensions	<ul style="list-style-type: none"> • ACE17—Confirm that file extensions exist in only one component.
Files	<ul style="list-style-type: none"> • ACE23—Identifies file duplication between source and target packages. • ACE12—Checks for components that contain files that can be replaced by one of the merge modules that you have imported into the Application Catalog database. • ACE08—identifies components with identical ComponentIds, and checks those components to see if the versions of the files in each component match. • ACE07—Checks for the existence of the same file in components with different ComponentIds. • ACE03—Checks to see if components in different packages that have matching ComponentIds also contain the same files.
INI Files	<ul style="list-style-type: none"> • ACE22—Checks file name/target directory pairs in the File table to see if they conflict with similar entries in the IniFile table. • ACE21—Checks entries in the IniFile table to see if they conflict with similar entries in the File table. • ACE14—Checks for the existence of components with different ComponentIds that modify the same INI file entry, such as the [File Name/Section/Key/Value] entry.
NT Services	<ul style="list-style-type: none"> • ACE16—Checks for the existence of identical NT Services in components with different ComponentIds.
ODBC Resources	<ul style="list-style-type: none"> • ACE15—Checks for the existence of identical ODBC entries in components with different ComponentIds.
Product Properties	<ul style="list-style-type: none"> • ACE20—Checks the Upgrade Code to see if it is unique. • ACE19—Checks the Product Code to see if it is unique. • ACE18—Checks the Package Code to see if it is unique.
Registry	<ul style="list-style-type: none"> • ACE24—Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value. • ACE10—Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.

Category	ACEs
Shortcuts	<ul style="list-style-type: none">• ACE13—Checks for the existence of the same shortcut within different packages in components with different ComponentIds.
Windows Terminal Server Validation	<ul style="list-style-type: none">• WST01—Checks to see if the ALLUSERS property is defined.• WST02—Checks for any registry entries installed to per-user specific locations.• WST03—Checks for any file entries installed to per-user specific locations.• WST04—Checks for any ODBC data source entries installed to per-user specific locations.• WST05—Checks for any environment strings installed to per-user specific locations.

5. Click **Save**.

Managing Jobs Using Job Manager Engine

Job Manager Engine executes the Jobs created with Job Manager. Job Manager Engine can connect to any of the Application Catalogs in your network and allow you to configure how the engine operates and manage the state of any particular Application Catalog.

This section includes the following topics:

- [About Job Manager Engine](#)
- [Opening the Job Manager Engine User Interface](#)
- [Adding an Application Catalog](#)
- [Deleting a Connection to an Application Catalog](#)
- [Executing Jobs](#)
- [Pausing, Shutting Down, and Restarting Application Catalog Jobs](#)
- [Setting Configuration Options](#)

About Job Manager Engine

The Job Manager Engine's **Manage Application Catalogs View** lists the **Catalog Name** of each of the Application Catalogs that it is connected to, the **Status** (Running, Pending, Paused) of each Application Catalog, and the **# of Jobs** from that Application Catalog that are currently running or waiting to run.

Opening the Job Manager Engine User Interface

In order to execute jobs, Job Manager Engine needs to be running somewhere in the network. Because Jobs are often scheduled to be run overnight, Job Manager Engine usually needs to be running continuously. There are two ways to launch Job Manager Engine:

- **Launch at startup**—The first time that Job Manager Engine launches, you are prompted to select whether you want to automatically launch Job Manager Engine at startup. If you choose this option, Job Manager Engine launches, but then is automatically minimized, and its icon is shown in the Windows system tray.
- **Select shortcut on Start menu**—You can open Job Manager Engine by selecting it from the **AdminStudio 9.5 Tools** subfolder on the **Start** menu. However, when it is launched, it is still automatically minimized, and its icon is shown in the Windows system tray.

If you want to view the Job Manager Engine interface, click on the Job Manager Engine system tray icon. The [Manage Application Catalogs View](#) of the Job Manager Engine opens.



For more information on the Job Manager Engine interface, see [Job Manager Engine](#).



Note • While everyone who installs AdminStudio Enterprise Server gets the Job Manager Engine, only one Job Manager Engine can be connected at any one time to any one Application Catalog.

Adding an Application Catalog

To add an Application Catalog to Job Manager Engine, perform the following steps.



Task: *To add an Application Catalog to Job Manager Engine:*

1. Open Job Manager Engine.
2. On the [Manage Application Catalogs View](#), click the **Add** button. The Select Application Catalog dialog box opens.

Deleting a Connection to an Application Catalog

When you delete an Application Catalog from Job Manager Engine, it will first shut down the Application Catalog, allowing any currently executing Job Steps to complete. Therefore, this operation may take some time to complete.



Task: *To delete the connection of Job Manager Engine to an Application Catalog*

1. Open Job Manager Engine.
2. On the [Manage Application Catalogs View](#), select the Application Catalog that you want to delete and click the **Delete** button. You are prompted to confirm the deletion.
3. Click **Yes**. The Application Catalog is no longer listed.

Executing Jobs

When Job Manager Engine opens, it automatically identifies all of the Jobs that are in **Pending** status. Then, as the scheduled time interval of each Pending Job is reached, Job Manager Engine executes it.

Job Manager Engine can be connected to multiple Application Catalogs at once, and multiple Job Manager Engines may run on the network. However an Application Catalog can only be connected to one Job Manager Engine at a time.

Pausing, Shutting Down, and Restarting Application Catalog Jobs

From the [Manage Application Catalogs View](#) of Job Manager Engine, you can Pause, Shutdown, and Restart Jobs in an Application Catalog:

Table 23-7 • Pausing, Shutting Down, and Restarting Application Catalog Jobs




Button	Action That is Taken by Job Manager Engine
Pause	<p>Click to pause the execution of the Jobs in the selected Application Catalog.</p> <p>Clicking the Pause button tells Job Manager Engine not to start the execution of any new Jobs in the selected Application Catalog, but allow all currently executing jobs to complete normally.</p> <p>Job Manager Engine remains fully aware of the Application Catalog, but does not use any CPU resources to run any Jobs. Therefore, the Pause operation could be used by someone to temporarily give another Application Catalog better access to system resources so that it could complete its jobs faster.</p> <p>When you select a Paused Application Catalog and click Run/Restart, its Pending Jobs begin executing immediately.</p>

Table 23-7 • Pausing, Shutting Down, and Restarting Application Catalog Jobs

Button	Action That is Taken by Job Manager Engine
Shutdown	<p>Click to stop all Jobs on the selected Application Catalog. Any currently running Job Step is allowed to finish, but no additional Jobs will execute.</p> <p>Clicking Shutdown tells Job Manager Engine to fail any currently running Jobs and to unload itself from memory. All resources used by Job Manager Engine for this Application Catalog are freed.</p> <p>When you select a Shutdown Application Catalog and click Run/Restart, the restart operation will need to relaunch and initiate the background thread for the Application Catalog, and therefore a Shutdown Application Catalog takes longer to restart than a Paused Application Catalog.</p>
Run/Restart	Click to initiate execution of Jobs from a selected Application Catalog that was Paused or Shutdown.

When you select an Application Catalog, an icon is displayed to indicate its status:

Table 23-8 • Job Manager Engine Status Icons

Icon	Status
	All Pending Jobs and any future Jobs sent from this Application Catalog will be executed.
	The Pending Jobs from this Application Catalog will not be executed until the connection is restarted by clicking the Run/Start button.
	<p>No Jobs from this Application Catalog will be executed until the Application Catalog is restarted by clicking the Run/Start button.</p> <p>All Jobs from this Application Catalog that were in Pending status when it was shutdown were Failed and must be resubmitted to execute.</p>

Viewing Job Manager Engine Status in Job Manager

The status (**Active** or **Not Active**) of the Job Manager Engine associated with an installation of Job Manager is displayed in the **Job Manager Engine Status** area of the left pane of all of the Job Manager pages:

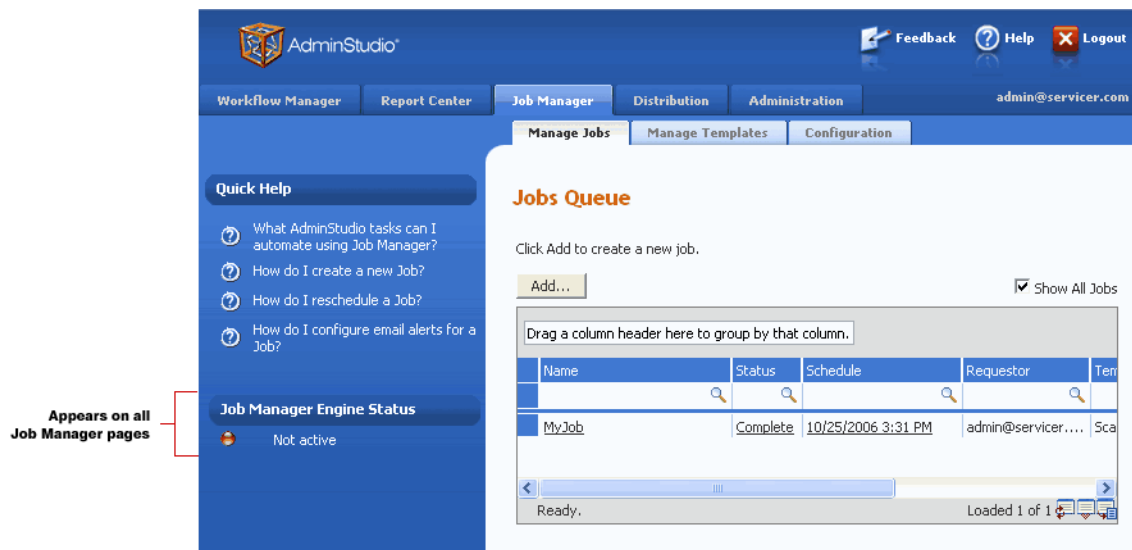


Figure 23-2: Job Manager Engine Status Displayed in the Job Manager User Interface

Setting Configuration Options

Usually the Job Manager Engine is installed on the same Server as the AdminStudio client tools, and the Job Manager Configuration options are set on the **Rules** tab and **Validate** tab of the ConflictSolver **Options** dialog box.

However, if you choose to install Job Manager Engine on a different server, you can set the following configuration options from the Job Manager Engine interface:

- [Conflict Detection Configuration](#)
- [Validation Configuration](#)
- [Email Alerts Configuration](#)

Conflict Detection Configuration

On the **Conflict Detection Configuration View**, you can set the following configuration options:

Table 23-9 • Conflict Detection View Options

Option	Description
ACE File	File containing the pre-defined ACE rule information for all ConflictSolver Conflict and Best Practice ACEs. It is installed in the following location on the machine where AdminStudio is installed: <i>AdminStudio Installation Directory \ConflictSolver\Support</i>

Table 23-9 • Conflict Detection View Options

Option	Description
Custom ACE File	<p>You use user-defined custom ACEs to extend the functionality of pre-defined ACEs with company-specific functionality. By selecting different user-defined ACE files, you can organize rules appropriate for individual users in your organization.</p> <p>The user-defined ACE file specified here is run after the pre-defined ACE rules are run. The selection of this user-defined ACE file will affect the default Conflict Types displayed on the Job Manager interface when you are creating Job Steps.</p> <p>By default, an empty user-defined ACE file is installed in the following location on the machine where AdminStudio is installed.</p> <p>C:\AdminStudio Shared\ConflictSolver\ CustomConflictFile.ace</p> <p>Only one user-defined ACE file can be active at one time.</p>

Validation Configuration

On the **Validation Configuration View**, you can set the following configuration options:

Table 23-10 • Validation View Options

Option	Description
CUB File	<p>Validation involves comparing a Windows Installer package against a known criteria to identify deviations from those rules. By default, ConflictSolver compares packages against the Full MSI Validation Suite. This suite contains a comprehensive set of Internal Consistency Evaluators (ICEs)—guidelines created by Microsoft to ensure an installation package works correctly with the Windows Installer engine.</p> <p>The file specified in this field is file containing Internal Consistency Evaluators (ICEs) used for validation.</p> <p>The CUB file is configured as a part of the Job Server Engine installation, so normally you do not need to modify this entry.</p>

Email Alerts Configuration

On the **E-mail Alerts View**, you can enter configuration settings so that email alerts can be sent out to report the status of a completed or failed job.

Table 23-11 • E-mail Alerts View Options

Option	Description
SMTP Mail Server Name	<p>Enter the SMTP mail server the Job Manager Server should use when sending messages. Enter the MTP Mail Server in the format of:</p> <p>mail.mycompany.com</p>

Table 23-11 • E-mail Alerts View Options

Option	Description
Senders E-mail Address	Enter the E-mail Address of the person who you want to be displayed as the email sender.
SMTP Server Port	<p>When someone sends a mail message to your SMTP Server, the mail will attempt to come in to your server on Port 25. This is the standard SMTP (Simple Mail Transport protocol) port.</p> <p>Port 2525 is an alternate port used by forward services. This is a non-standard port, but useful if the standard SMTP port is blocked</p>

Job Manager Reference

Job Manager Reference includes the same topics displayed when you select help from the Job Manager or Job Manager Engine interface. Reference information for Job Manager is organized into the following areas:

Table 23-12 • Job Manager Reference Organization

Page	Views
Jobs Queue Page	<ul style="list-style-type: none"> • Add Job Page • Edit Job Page • Delete Job/Template Page • Schedule Job Page • Job Alerts Page • Job Status Page
Job Template List Page	<ul style="list-style-type: none"> • Add Template Page • Edit Template Page
Conflict Detection Options Page	
Job Manager Engine	<ul style="list-style-type: none"> • Manage Application Catalogs View • Conflict Detection Configuration View • Validation Configuration View • E-mail Alerts View

Jobs Queue Page

The **Jobs Queue** page lists all defined Jobs. From this page you can choose to add a new Job, edit an existing Job, delete a Job, schedule a job, or define the email alerts that will be sent out when this job is executed.

The following information and options are available:

Table 23-13 • Jobs Queue Page Options

Option	Description
Add Button	Click to open the Add Job Page where you can create a new Job.
Name	Name of Job
Status	<p>Current status of the Job in the Job Execution process:</p> <ul style="list-style-type: none">• Scheduled—Job is scheduled to be executed at a future date and time.• Running—Job is currently being run. Only one job per Application Catalog connection will ever be running at one time.• Pending—Job is available to run but is waiting for the running job to complete. A job set for immediate execution will be automatically assigned a Pending job status.• Complete—Job has been executed.• Canceled—Job was canceled before it was executed or before its execution was complete.• Failed—Job was unable to run either because of a software error, a network communication problem, a hardware failure, or any other reason. <p>The values in the Status column are also hyperlinks. Click on a status value to open the Job Status Page that lists the current status of each Job Step in this Job.</p>
Schedule	Lists the scheduled date and time for this job. Click this link to open the Schedule Job Page where you can specify when to run this job and also configure the selected Job to reoccur over a specified interval.
Requestor	User who created this Job.
Template	Template that was used to create the Job. If no Template is listed, it is a Custom Job.
Alerts	Click the link in this column to open the Job Alerts Page where you can add or edit an email to send when the selected Job is completed and when it fails.
Action	Click Delete in this column for a Job you want to delete.

Add Job Page

The **Add Job** page includes the [Template View](#) and the [Custom Job View](#).

Template View

On the **Job Template** View, you create a new Job based on a pre-existing Job Template. The **Job Template** View contains the following options:

Table 23-14 • Add Job Page / Job Template View Options

Option	Description
Job Name	Enter a name to identify the Job.
Template	Select this option and select a Template from the list to create a Template Job. .
Package List	Select the package(s) or group(s) to be used by the Job Template. If you select a group, the Job will perform actions on whatever packages are in the selected group at <i>the time the Job is executed</i> .
Execution	Use the following options to schedule this Job: <ul style="list-style-type: none"> • Immediate—Select to execute this Job immediately. • Scheduled—Select to execute this Job at the Date and Time you select.

Custom Job View

The Custom Job View contains the following options:

Table 23-15 • Add Job Page / Custom Job View Options

Option	Description
Name	Enter a name to identify the Job.
Template	To create a Custom Job, clear this selection. When this option is not selected, the Define Job Steps area opens.

Table 23-15 • Add Job Page / Custom Job View Options (cont.)

Option	Description
Define Job Steps	<p>You create Job Steps using this area. To create the first Job Step, make a selection from the Step Type list and click Next. To create all subsequent Job Steps, click Insert New Step.</p> <p>The following Job Step Types are available:</p> <ul style="list-style-type: none"> ● Import—Import a selected package into the Application Catalog. ● Scan for Dependencies—Analyze a Windows Installer package for dependent files. ● Directory Monitoring—Use to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files, and if any changes are detected, those new or modified files are then either imported or re-imported into the Application Catalog. See About Automatically Importing Packages from a Network Directory for more information. ● Validation—Verifies the internal data integrity of a Windows Installer package. ● Conflict Detection—Performs conflict detection on selected packages using Conflict ACE Rules ● Best Practice Detection—Performs conflict detection on a single package using Best Practice ACE Rules to enforce Windows Installer standards. ● Resolution—Performs automatic conflict resolution on selected packages. <p>For each type of Job Step, you are prompted to specify additional information. See Information Requested When Defining Job Steps in a Custom Job.</p> <p>Use the Up and Down arrows to rearrange the order of the Job Steps, and use the Delete button to delete a Job Step.</p> <p>Click Finish to save each Step and click Save to save the Job.</p>
Schedule Job	<p>Use the following options to schedule this Job:</p> <ul style="list-style-type: none"> ● Execute—Select to execute this job Immediately or at a Scheduled time. ● Select Date—If you chose to execute this job at a Scheduled time, click the calendar icon and select a date. ● Enter Time—If you chose to execute this job at a Scheduled time, enter the time in hours and minutes.

Information Requested When Defining Job Steps in a Custom Job

Each time you select a **Step Type**, you are prompted to specify additional information:

Table 23-16 • Information Requested When Defining a Job Step in a Custom Job

Step Type	Required Selections
Scan for Dependencies	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. Select the packages or groups of packages that you want to scan, and then click Finish to save this Job Step.
Directory Monitoring	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. In the Directory Name box, enter the UNC path of the directory that you want to monitor. 3. Choose whether to Include subdirectories in the 'Directory Monitoring' process. 4. Click Finish to save this Job Step.
Validation	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 2 View opens. 2. Select the packages or groups of packages that you want to validate. 3. Click Finish to save this Job Step.
Conflict Detection	<ol style="list-style-type: none"> 1. Click Next. The Step 2 of 4 View opens. 2. Select the source packages or groups of packages that you want to include in the conflict detection process, and then click Next. The Step 3 of 4 View opens. 3. Select the target packages or groups of packages that you want to use in the conflict detection process and click Next. The Step 4 of 4 View opens. 4. Select the set of Conflict ACE Rules that you want to include in the conflict detection process and click Finish to save this Job Step.
Import	<ol style="list-style-type: none"> 1. Click Next. the Step 2 of 4 View opens. 2. Select a package to import. 3. Specify whether to include the package in the Software Repository 4. Click Next. The Step 3 of 4 View opens. 5. Select any transform (.mst) or patch (.msp) files to import. 6. Click Next. The Step 4 of 4 View opens. 7. Select one or more destination groups into which the package will be imported. 8. Click Finish to save this Job Step.

Table 23-16 • Information Requested When Defining a Job Step in a Custom Job (cont.)

Step Type	Required Selections
Best Practice Detection	<ol style="list-style-type: none">1. Click Next. The Step 2 of 2 View opens.2. Select the packages or groups of packages that you want to perform Best Practice conflict detection on.3. Click Finish to save this Job Step.
Resolution	<ol style="list-style-type: none">1. Click Next. The Step 2 of 3 View opens.2. Select the packages or groups of packages that you want to perform conflict resolution on, and then click Next. The Step 3 of 3 View opens.3. Select the Conflict ACE Rules that you want to resolve automatically.4. Select a Resolved packages re-import option. See Re-Import Behaviors.5. Click Finish to save this Job Step.

Edit Job Page

You can edit an existing Job by selecting the Job on the **Jobs Queue** page to open the **Edit Job** page.



Note • When editing Jobs note that:

- Only one Job can be edited at a time.
- You cannot edit a Job with a **Status** of **Running**.
- You cannot change a **Custom Job** to a **Template Job** or vice versa.

You edit a Job following the instructions in [Creating a New Job Based on a Template](#) or [Creating a New Custom Job](#). When you have finished editing the Job, click **Save**.

Editable Items in an Existing Job

The following information can be edited:

Table 23-17 • Editable Items in an Existing Job

Custom Jobs	Template Jobs
<p>General</p> <ul style="list-style-type: none"> • Job Name • Execution options • Step Type <p>Scan for Dependencies Job Step</p> <ul style="list-style-type: none"> • Selected packages and groups <p>Directory Monitoring Job Step</p> <ul style="list-style-type: none"> • Directory to monitor • Option to monitor subdirectories <p>Validation Job Step</p> <ul style="list-style-type: none"> • Selected packages and groups <p>Conflict Detection Job Step</p> <ul style="list-style-type: none"> • Selected source packages and groups • Selected target packages and groups • Set of conflict detection rules to use during the conflict detection process <p>Import Job Step</p> <ul style="list-style-type: none"> • Package to import • Software Repository option • Transform (.mst) and Patch (.msp) files to include during import • Destination group for imported package <p>Best Practice Detection Job Step</p> <ul style="list-style-type: none"> • Selected packages and groups <p>Resolution Job Step</p> <ul style="list-style-type: none"> • Selected packages and groups • Conflicts to resolve • Resolved package re-import options 	<p>General</p> <ul style="list-style-type: none"> • Job Name • Execution options <p>General</p> <ul style="list-style-type: none"> • Selected packages and groups

Delete Job/Template Page

The **Delete Job/Template** page opens when you attempt to delete a Job or Template and prompts you to confirm the deletion. To delete a Job or Template:

- **Deleting a Job**—Open the **Jobs Queue** page and click **Delete** in the **Action** column of a Job that you want to delete.
- **Deleting a Template**—Open the **Job Template List** page and click **Delete** in the **Action** column of a Template that you want to delete.

To confirm the deletion, click **Delete**. To return to the **Jobs Queue** or **Job Template List** page without making the deletion, click **Cancel**.

Schedule Job Page

You can reschedule a completed Job so that it is run again. You can also reschedule Jobs that failed, were cancelled, or are scheduled to be run in the future.

To reschedule a Job, open the **Jobs Queue** page and click the date and time listed in the **Schedule** column for the Job you want to reschedule. The **Schedule Job page** opens.

The **Schedule Job** page includes the following options:

Table 23-18 • Schedule Job Page Options

Option	Description
Execute	Select For Immediate Execution or For Scheduled Execution .
Select Date	If you selected For Scheduled Execution , click the calendar icon and select the date when you want this Job to run.
Enter Time	If you selected For Scheduled Execution , identify the time of day that you want this Job to run by selecting an hour of the day from the first list, selecting the minutes after the hour from the second list, and selecting AM or PM from the third list.
Recurrence Interval	If you want this Job to reoccur at a specified interval, select Weekly, Daily, or Monthly from this list, and perform the following steps: <ul style="list-style-type: none">• If you want this Job to reoccur a specific number of times, select the Number of Times option, and enter a number in the box.• If you want this Job to reoccur until a specified expiration date, select the Expiration Date option, click the calendar icon, and select a date from the calendar.
Save	Click to save this Job schedule.

Job Alerts Page

You can choose to send an email when a Job is completed and when it fails. To do this, open the **Jobs Queue** page and click **Add** in the **Alerts** column of the Job that you want to set an alert for. You then can create an email message on the **Job Alerts** page.

When composing the **Subject** line and the **Message**, you can insert system variables to instruct Job Manager to automatically insert the variable data regarding that Job at runtime.

The **Job Alerts** page includes the following options:

Table 23-19 • Job Alerts Page Options

Options	Description
To	Enter the email addresses that you want to be notified when a Job is completed or when it fails. Separate multiple email addresses with a semicolon (;).
Subject	Enter text to describe the purpose of the email.
Message	Enter the text of the email message. You can select System Variables from the list so that the value of the variable for this Job is inserted when the email is sent.
Conditional Forward	Select one of the following options: <ul style="list-style-type: none"> • Failed—Send an email when this Job fails. • Complete—Send an email when this Job is completed.
Save	Click to save your entries.

Job Status Page

On the **Jobs Queue** page, a Job's current status is listed as a hyperlink in the **Status** column. If you click on this hyperlink, the **Job Status** page opens, displaying the **Job Name** and **Job Status**, and the status of each of the Job Steps in that job.

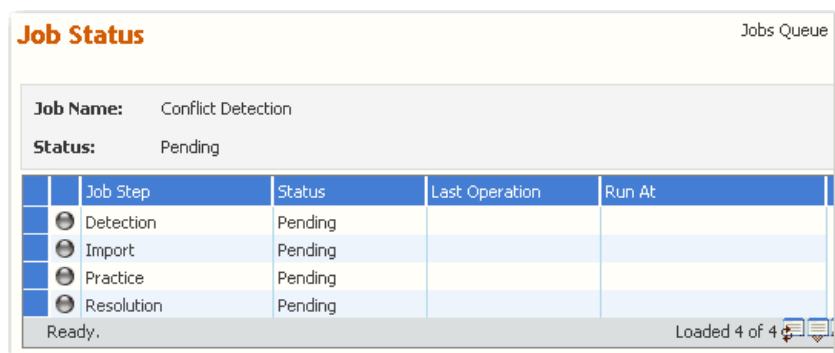


Figure 23-3: Job Status Page

The possible statuses are:

- **Scheduled**—Job is scheduled to be executed at a future date and time.
- **Running**—Job is currently being run. Only one job per Application Catalog connection will ever be running at one time.
- **Pending**—Job is available to run but is waiting for the running job to complete. A job set for immediate execution will be automatically assigned a **Pending** job status.
- **Complete**—Job has been executed.
- **Canceled**—Job was canceled before it was executed or before its execution was complete.
- **Failed**—Job was unable to run either because of a software error, a network communication problem, a hardware failure, or any other reason.

Click **Back** to return to the **Jobs Queue** page.

Job Template List Page

The **Job Template List** page lists all defined Templates. From this page you can choose to add a new Template, edit an existing Template, or delete a Template.

The following information and options are available:

Table 23-20 • Job Template List Page Options

Option	Description
Add	Click to open the Add Template page where you can create a new Template.
Template Name	Name of Template
# of Steps	Number of Job Steps defined in the Template.
Description	Description that was entered to identify the Template.
Action	Click Delete in this column to delete a listed Template.

Add Template Page

The **Add Template** page contains the following options:

Table 23-21 • Add Template Page Options

Option	Description
Name	Enter a name to identify the new Template.

Table 23-21 • Add Template Page Options (cont.)

Option	Description
Description	Enter a description to identify the purpose of this Template.
Define Job Steps	<p>You create Job Steps using this area. Each time you want to create a Job Step, select one of the listed tasks from the Step Type list:</p> <ul style="list-style-type: none"> • Scan for Dependencies—Analyze a Windows Installer package for dependent files. • Directory Monitoring—Use to establish a consistent, repeatable process in which a directory and, optionally, its subdirectories are monitored for any new or modified Windows Installer package files, and if any changes are detected, those new or modified files are then either imported or re-imported into the Application Catalog. See About Automatically Importing Packages from a Network Directory for more information. • Validation—Verifies the internal data integrity of a Windows Installer package. • Conflict Detection—Performs conflict detection on selected packages using Conflict ACE Rules • Best Practice Detection—Performs conflict detection on a single package using Best Practice ACE Rules to enforce Windows Installer standards. • Resolution—Performs automatic conflict resolution on selected packages. <p>For each type of Job Step, you are prompted to specify additional information. See Information Requested When Defining Job Steps in a Template.</p> <p>Use the Up and Down arrows to rearrange the order of the Job Steps, and use the Delete button to delete a Job Step.</p> <p>Click Finish to save each Step and click Save to save the Job.</p>

Information Requested When Defining Job Steps in a Template

Each time you select a **Step Type**, you are prompted to specify additional information:

Table 23-22 • Information Requested When Defining a Job Step in a Template

Step Type	Required Selections
Scan for Dependencies	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)

Table 23-22 • Information Requested When Defining a Job Step in a Template (cont.)

Step Type	Required Selections
Directory Monitoring	<ol style="list-style-type: none">1. Click Next. The Step 2 of 2 view opens.2. In the Directory Name box, enter the UNC path of the directory that you want to monitor.3. Choose whether to Include subdirectories in the ‘Directory Monitoring’ process.4. Click Finish to save this Job Step.
Validation	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)
Conflict Detection	<ol style="list-style-type: none">1. Click Next. The Step 2 of 3 view opens.2. Select the target packages or groups of packages that you want to use in the conflict detection process and click Next. The Step 3 of 3 view opens.3. Select the set of Conflict ACE Rules that you want to include in the conflict detection process and click Finish to save this Job Step.
Best Practice Detection	Click Next to save this Job Step. (Source packages will be selected when a Job based on this Template is created.)
Resolution	<ol style="list-style-type: none">1. Click Next. The Step 2 of 2 view opens.2. Select the Conflict ACE Rules that you want to resolve automatically.3. Select a Resolved packages re-import option. See Re-Import Behaviors.4. Click Finish to save this Job Step.

Edit Template Page

You can edit an existing Template by selecting that Template on the **Job Template List** page to open the **Edit Template** page.

Edit the Template following the instructions in [Creating a New Template](#). When you have finished editing the Template, click **Save**.

Conflict Detection Options Page

On the **Conflict Detection Options** page, you select the rules that you want Job Manager to use by default during conflict detection. The following rules are available.

Table 23-23 • Conflict Detection Options

Category	ACEs
Components	<ul style="list-style-type: none"> • ACE30—A check for KeyPath conflicts across components. • ACE02—Check destination directories for identical components. • ACE09—Check for different merge module versions.
File Extensions	<ul style="list-style-type: none"> • ACE17—Confirm that file extensions exist in only one component.
Files	<ul style="list-style-type: none"> • ACE23—Identifies file duplication between source and target packages. • ACE12—Checks for components that contain files that can be replaced by one of the merge modules that you have imported into the Application Catalog database. • ACE08—identifies components with identical ComponentIds, and checks those components to see if the versions of the files in each component match. • ACE07—Checks for the existence of the same file in components with different ComponentIds. • ACE03—Checks to see if components in different packages that have matching ComponentIds also contain the same files.
INI Files	<ul style="list-style-type: none"> • ACE22—Checks file name/target directory pairs in the File table to see if they conflict with similar entries in the IniFile table. • ACE21—Checks entries in the IniFile table to see if they conflict with similar entries in the File table. • ACE14—Checks for the existence of components with different ComponentIds that modify the same INI file entry, such as the [File Name/Section/Key/Value] entry.
NT Services	<ul style="list-style-type: none"> • ACE16—Checks for the existence of identical NT Services in components with different ComponentIds.
ODBC Resources	<ul style="list-style-type: none"> • ACE15—Checks for the existence of identical ODBC entries in components with different ComponentIds.
Product Properties	<ul style="list-style-type: none"> • ACE20—Checks the Upgrade Code to see if it is unique. • ACE19—Checks the Product Code to see if it is unique. • ACE18—Checks the Package Code to see if it is unique.

Table 23-23 • Conflict Detection Options (cont.)

Category	ACEs
Registry	<ul style="list-style-type: none">• ACE24—Checks to see if registry entries with the same registry hive, key, and value name have the same data type and value.• ACE10—Checks for the existence of identical Root/Key/Name registry combinations in components with different ComponentIds.
Shortcuts	<ul style="list-style-type: none">• ACE13—Checks for the existence of the same shortcut within different packages in components with different ComponentIds.
Windows Terminal Server Validation	<ul style="list-style-type: none">• WST01—Checks to see if the ALLUSERS property is defined.• WST02—Checks for any registry entries installed to per-user specific locations.• WST03—Checks for any file entries installed to per-user specific locations.• WST04—Checks for any ODBC data source entries installed to per-user specific locations.• WST05—Checks for any environment strings installed to per-user specific locations.

Job Manager Engine

Job Manager Engine executes the Jobs created with Job Manager. Job Manager Engine can connect to any of the Application Catalogs in your network and allow you to configure how the engine operates and manage the state of any particular Application Catalog.

Job Manager Engine has the following Views:

- [Manage Application Catalogs View](#)
- [Conflict Detection Configuration View](#)
- [Validation Configuration View](#)
- [E-mail Alerts View](#)

Manage Application Catalogs View

Application Catalogs that are connected to the Job Manager Engine are listed on the Manage Application Catalogs View. For each Application Catalog, the following information is listed:

Table 23-24 • Connected Application Catalog Information

Item	Description
Catalog Name	Name of the connected Application Catalog database.
Status	Identifies whether the Job is Running, Pending, or Shutdown.


Table 23-24 • Connected Application Catalog Information (cont.)

Item	Description
# of Jobs	Number of Jobs in that Application Catalog that have been submitted for execution.

When Job Manager Engine is open, it automatically identifies all of the Jobs in all of the Application Catalogs that it is connected to that are in **Pending** status. Then, as the scheduled time interval of each Pending Job is reached, Job Manager Engine will execute it.

On the Manage Application Catalogs View, the following functions can be performed:

Table 23-25 • Job Manager Engine / Manage Application Catalogs View Options

Option	Description
Delete	Click to delete the connection of Job Manager Engine to an Application Catalog.  <i>Note • When you delete an Application Catalog from Job Manager Engine, it will first shut down the Application Catalog, allowing any currently executing Job Steps to complete. Therefore, this operation may take some time to complete.</i>
Run/Restart	Click to initiate execution of Jobs from a selected Application Catalog that was Paused or Shutdown.
Pause	Click to pause the execution of the Jobs in the selected Application Catalog.
Shutdown	Click to stop all Jobs on the selected Application Catalog. Any currently running Job Step is allowed to finish, but no additional Jobs would execute.

When you select an Application Catalog on Job Manager Engine Manage Application Catalogs View, an icon is displayed to indicate its status:

Table 23-26 • Job Manager Engine Status Icons




Icon	Status
	All Pending Jobs and any future Jobs sent from this Application Catalog will be executed.
	The Pending Jobs from this Application Catalog will not be executed until the connection is restarted by clicking the Run/Start button.

Table 23-26 • Job Manager Engine Status Icons (cont.)

Icon	Status
	<p>No Jobs from this Application Catalog will be executed until the Application Catalog is restarted by clicking the Run/Start button.</p> <p>All Jobs from this Application Catalog that were in Pending status when it was shutdown were Failed and must be resubmitted to execute.</p>

Conflict Detection Configuration View

On the Conflict Detection Configuration View, you can set the following configuration options:

Table 23-27 • Conflict Detection Configuration View Options

Option	Description
ACE File	<p>File containing the pre-defined ACE rule information for all ConflictSolver Conflict and Best Practice ACEs. It is installed in the following location on the machine where AdminStudio is installed:</p> <p><i>AdminStudio Installation Directory \ConflictSolver\Support</i></p>
Custom ACE File	<p>You use user-defined custom ACEs to extend the functionality of pre-defined ACEs with company-specific functionality. By selecting different user-defined ACE files, you can organize rules appropriate for individual users in your organization.</p> <p>The user-defined ACE file specified here is run after the pre-defined ACE rules are run. The selection of this user-defined ACE file will affect the default Conflict Types displayed on the Job Manager interface when you are creating Job Steps.</p> <p>By default, an empty user-defined ACE file is installed in the following location on the machine where AdminStudio is installed.</p> <p>C:\AdminStudio Shared\ConflictSolver\ CustomConflictFile.ace</p> <p>Only one user-defined ACE file can be active at one time.</p>

Validation Configuration View

On the Validation Configuration View, you can set the following configuration options:

Table 23-28 • Validation Configuration View Options

Option	Description
CUB File	<p>Validation involves comparing a Windows Installer package against a known criteria to identify deviations from those rules. By default, ConflictSolver compares packages against the Full MSI Validation Suite. This suite contains a comprehensive set of Internal Consistency Evaluators (ICEs)—guidelines created by Microsoft to ensure an installation package works correctly with the Windows Installer engine.</p> <p>The file specified in this field is file containing Internal Consistency Evaluators (ICEs) used for validation.</p> <p>The CUB file is configured as a part of the Job Server Engine installation, so normally you do not need to modify this entry.</p>

E-mail Alerts View

On the E-mail Alerts View, you can enter configuration settings so that email alerts can be sent out to report the status of a completed or failed job.

Table 23-29 • E-mail Alerts View Options

Option	Description
SMTP Mail Server Name	<p>Enter the SMTP mail server the Job Manager Server should use when sending messages. Enter the MTP Mail Server in the format of:</p> <p>mail.mycompany.com</p>
Senders E-mail Address	<p>Enter the E-mail Address of the person who you want to be displayed as the email sender.</p>
SMTP Server Port	<p>When someone sends a mail message to your SMTP Server, the mail will attempt to come in to your server on Port 25. This is the standard SMTP (Simple Mail Transport protocol) port.</p> <p>Port 2525 is an alternate port used by forward services. This is a non-standard port, but useful if the standard SMTP port is blocked</p>

AdminStudio.Public API Reference

The AdminStudio.Public API Reference help can be viewed online as part of the AdminStudio Client Tools Help Library. You can also view the AdminStudio.Public API Reference on HelpNet at:

<http://helpnet.flexerasoftware.com/Robo/BIN/Robo.dll?tpc=/robo/projects/as90api/html/ASAPIIntro.htm>



Note • The AdminStudio.Public API Reference help is also included as a separate help file on the AdminStudio client tools installation DVD: *AdminStudio.Public Reference.chm*.

Index

Symbols

-? [426](#), [516](#)
.aot [527](#)
 converting to Repackager project [527](#)
 difference from .axt file [528](#)
.axt [527](#)
 converting to Repackager project [527](#)
 difference from .aot file [528](#)
.cab [1027](#)
.cer file [1358](#)
.cub [1016](#), [1020](#)
.inc [516](#), [527](#), [628](#), [635](#)
 converting to Repackager project [527](#)
.ini Files Tab [624](#)
.ipf [527](#)
 converting to Repackager project [527](#)
.irp [525](#), [635](#)
 creating [525](#)
.isl [531](#)
 converting to Repackager project [531](#)
.ism [129](#), [378](#), [516](#), [532](#), [628](#)
 building in Repackager [532](#)
.msi [532](#)
 building in Repackager [532](#)
.osc [302](#)
.pvk file [1359](#)
.spc file [1359](#)
.spy [635](#)
.txt [531](#)
 converting to Repackager project [531](#)
.wse [531](#)
 converting to Repackager project [531](#)

Numerics

-10000 - Process Cancelled By User [756](#)
-10001 - Suite File Missing [757](#)
-10002 - Suite File is Duplicate [757](#)
-10003 - Application File Missing [757](#)
-10004 - INI File Missing [758](#)
11000 - Excluding TCPIP Registry Entries [758](#)
11001 - Fail on VMware [759](#)
11003 - Control Panel Applet - Citrix [759](#)
11004 - Control Panel Applet - ThinApp [759](#)
11005 - QuickTime 7.4.1 Causes Fatal Error [760](#)
11006 - Adobe Distiller Exclude AdobePDFSettings [760](#)
11007 - Exclude URL Shortcut [760](#)
64-bit applications [446](#)
-9000 - Unknown Exception [708](#)
-9001 - Unknown COM [708](#)
-9002 - Error Opening Package [709](#)
-9003 - Error Saving Package [709](#)
-9004 - Process Cancelled By User [710](#)
-9005 - Error Creating Temporary Folder [710](#)
-9006 - Error Decompressing Package [710](#)
-9007 - File With Extension Not Found [711](#)
-9008 - Error Extracting Icon [711](#)
-9009 - Unknown Provider [712](#)
-9010 - Invalid Target File Name [712](#)
-9011 - Error Reading MSI Table [713](#)
-9012 - Unexpected Error in Method [713](#)
-9013 - Type Library Not Found [713](#)
-9014 - ShellExecute Failed [714](#)
-9015 - Unable to Determine Full Path for Driver [714](#)
-9016 - Contents of Table Ignored [715](#)
-9017 - .NET 1.x Assembly Not Supported [716](#)

- 9018 - Custom Actions Warning [716](#)
- 9019 - Conditionalized Components [717](#)
- 9020 - Directory With Null Parent Error [718](#)
- 9021 - Unable to Extract COM Data [718](#)
- 9022 - Complus Table Error [719](#)
- 9024 - FileSFPCatalog [719](#)
- 9026 - LaunchCondition Table Warning [720](#)
- 9027 - LockPermissions Table Warning [720](#)
- 9028 - MoveFile Table Error [721](#)
- 9029 - MsiDriverPackages Table Error [722](#)
- 9030 - ODBCTranslator Table Warning [722](#)
- 9031 - RemoveFile Table Warning [723](#)
- 9032 - RemoveIniFile Table Warning [723](#)
- 9033 - RemoveRegistry Table Warning [724](#)
- 9036 - ISCEInstall Table Error [724](#)
- 9037 - ISComPlusApplication Table Error [725](#)
- 9038 - ISPalmApp Table Error [725](#)
- 9039 - ISSQLScriptFile Table Error [726](#)
- 9040 - ISVRoot Table Error [726](#)
- 9041 - ISXmlFile Table Error [727](#)
- 9051 - Package Decompression Canceled [727](#)
- 9100 - CreateInstance of Package Object Failed [727](#)
- 9101 - Create Operation of Package Object Failed [728](#)
- 9102 - Failed to Write Header Information [728](#)
- 9103 - Citrix Finalization Failed [729](#)
- 9104 - Citrix Save Failed [729](#)
- 9105 - Error Initializing Citrix Writer [729](#)
- 9106 - Error Initializing Citrix Package [730](#)
- 9107 - Error Writing Citrix File Entries [730](#)
- 9108 - Error Determining Source File Path [731](#)
- 9109 - Error Writing Citrix Folder Entries [731](#)
- 9110 - Error Writing Citrix Registry Entries [731](#)
- 9113 - Error Writing Citrix INI File Entries [732](#)
- 9114 - Error Writing Citrix Shortcuts [732](#)
- 9115 - Error Saving Citrix Profile [733](#)
- 9116 - Error Creating Empty Citrix Profile [733](#)
- 9117 - Error Creating Intermediate Folder [733](#)
- 9118 - Error Initializing Citrix Profile [734](#)
- 9119 - Error Creating Default Target in Citrix Profile [734](#)
- 9120 - Error Deleting File From Profile [734](#)
- 9121 - Failed to Copy File into Citrix Profile [735](#)
- 9122 - Target Does Not Exist in Citrix Profile [735](#)
- 9124 - No Shortcuts Created for this Profile [736](#)
- 9125 - Error Writing Citrix File Type Associations [736](#)
- 9126 - Failed to Sign Profile Using Certificate [737](#)
- 9127 - Could Not Create Script Execution [737](#)
- 9128 - Duplicate Shortcut [737](#)
- 9129 - Duplicate Shortcut Names [738](#)
- 9130 - Duplicate Shortcut Targets [738](#)
- 9131 - Unable to Resolve Installer Variable [739](#)
- 9132 - 16 Color Shortcut Icon Not Found [739](#)
- 9133 - Shortcut Icon Not Found [739](#)
- 9134 - Failure to Extract Icon from Executable [740](#)
- 9135 - Shortcut Target is 16-Bit [740](#)
- 9136 - Some Files May Not Be Decompressed [741](#)
- 9137 - Destination Directory Cannot Be Found [741](#)
- 9138 - DuplicateFile table warning [742](#)
- 9200 - ThinApp Must Be Installed [742](#)
- 9201 - Extension for Shortcut Files Must Be .exe [743](#)
- 9202 - No Applications Were Created [743](#)
- 9203 - ThinApp Tool is Missing [744](#)
- 9204 - Duplicate Shortcut [744](#)
- 9205 - Identically-Named Shortcut Already Exists, But With Different Command Line Parameters [744](#)
- 9206 - Identically-Named Shortcut Already Exists But With a Different Target [745](#)
- 9207 - Error During Build Process (vregtool.exe) [745](#)
- 9208 - Error Occurred During Build Process (vftool.exe) [746](#)
- 9209 - Error Occurred During Build Process (tlink.exe) [746](#)
- 9300 - Unhandled Exception During AdviseFile Operation [746](#)
- 9301 - Unhandled Exception During AdviseRegistry Operation [747](#)
- 9302 - Unhandled Exception During Command Action [747](#)
- 9303 - Unhandled Exception During Alter File Action [748](#)
- 9304 - Unhandled Exception During Alter Registry Action [748](#)
- 9305 - Unhandled Exception During Create Action [748](#)
- 9306 - Unhandled Exception During Execution of Rules Engine [749](#)
- 9401 - Error Initializing App-V Writer [749](#)
- 9402 - Error Initializing App-V Package [749](#)
- 9403 - Error Writing App-V File Entries [750](#)
- 9404 - Error Writing App-V Folder Entries [750](#)
- 9405 - Error Writing App-V Registry Entries [750](#)
- 9406 - Error Writing App-V INI File Entries [751](#)
- 9407 - Error Writing App-V Shortcuts [751](#)
- 9408 - Error Writing App-V File Type Data [751](#)
- 9409 - Error Saving App-V Data [752](#)
- 9410 - Error Determining Source File Path [752](#)
- 9411 - OSD File Template Could Not Be Extracted [752](#)
- 9412 - OSD File Could Not Be Saved [753](#)
- 9413 - App-V OSD Real Save [753](#)
- 9414 - Local App-V Application Should Not Be Specified as a Dependency of the Primary Application [753](#)
- 9415 - Dependency Application Was Not Found [754](#)
- 9416 - Invalid Primary Application Directory [754](#)
- 9417 - Dependency Application's OSD File Contains an Invalid HREF Value [755](#)
- 9418 - Error While Privatizing Side-By-Side Assemblies [755](#)
- 9419 - Error Inserting Watermark [755](#)
- 9500 - Shortcut Missing [756](#)

A

- About AdminStudio dialog [143](#)
- About Exclusions Editor dialog box [628](#)
- About InstallShield ConflictSolver dialog [1227](#)
- About QualityMonitor dialog [1409](#)
- About Repackager dialog box [587](#)
- access accounts [1510](#)
- accounts
 - deleting a group account [162](#)
 - deleting a user account [162](#)
- ACE Rule Properties dialog [1227](#), [1228](#), [1229](#), [1230](#), [1231](#)
 - Additional Information tab [1229](#), [1230](#)
 - DLL Information tab [1231](#)
 - General Information tab [1228](#)
 - Where Clause tab [1230](#)
- ACE02 [1118](#), [1146](#)
- ACE03 [1118](#), [1146](#)
- ACE04 [1118](#), [1147](#)
- ACE05 [1148](#)
- ACE06 [1118](#), [1149](#)
- ACE07 [1118](#), [1149](#), [1601](#)
- ACE08 [1118](#), [1151](#)
- ACE09 [1118](#), [1152](#)
- ACE10 [1118](#), [1153](#), [1601](#)
- ACE12 [1118](#), [1155](#)
- ACE13 [1118](#), [1157](#)
- ACE14 [1118](#), [1158](#)
- ACE15 [1118](#), [1159](#)
- ACE16 [1118](#), [1160](#)
- ACE17 [1118](#), [1161](#)
- ACE18 [1118](#), [1162](#)
- ACE19 [1118](#), [1163](#)
- ACE20 [1118](#), [1163](#)
- ACE21 [1118](#), [1164](#)
- ACE22 [1118](#), [1165](#)
- ACE23 [1118](#), [1167](#), [1601](#)
- ACE24 [1118](#), [1168](#), [1601](#)
- ACE25 [1170](#)
- ACE26 [1170](#)
- ACE27 [1171](#)
- ACE28 [1172](#)
- ACE29 [1173](#)
- ACE30 [1173](#)
- ACE31 [1174](#)
- ACE32 [1175](#)
- ACE33 [1176](#)
- ACE34 [1176](#)
- ACE35 [1177](#)
- ACE36 [1178](#)
- ACEs [1141](#), [1142](#), [1146](#), [1147](#), [1148](#), [1149](#), [1151](#), [1152](#), [1153](#), [1155](#), [1157](#), [1158](#), [1159](#), [1160](#), [1161](#), [1162](#), [1163](#), [1164](#), [1165](#), [1167](#), [1168](#), [1183](#), [1184](#), [1186](#), [1189](#), [1191](#), [1192](#), [1194](#)
- ACE02 [1146](#)
- ACE03 [1146](#)
- ACE04 [1147](#)
- ACE05 [1148](#)
- ACE06 [1149](#)
- ACE07 [1149](#)
- ACE08 [1151](#)
- ACE09 [1152](#)
- ACE10 [1153](#)
- ACE12 [1155](#)
- ACE13 [1157](#)
- ACE14 [1158](#)
- ACE15 [1159](#)
- ACE16 [1160](#)
- ACE17 [1161](#)
- ACE18 [1162](#)
- ACE19 [1163](#)
- ACE20 [1163](#)
- ACE21 [1164](#)
- ACE22 [1165](#)
- ACE23 [1167](#)
- ACE24 [1168](#)
- ACE25 [1170](#)
- ACE26 [1170](#)
- ACE27 [1171](#)
- ACE28 [1172](#)
- ACE29 [1173](#)
- ACE30 [1173](#)
- ACE31 [1174](#)
- ACE32 [1175](#)
- ACE33 [1176](#)
- ACE34 [1176](#)
- ACE35 [1177](#)
- ACE36 [1178](#)
- best practice ACEs [95](#)
- best practices ACEs [1171](#), [1174](#), [1176](#), [1177](#)
- creating custom [1184](#), [1186](#)
- creating DLL-based [1189](#)
- custom [1183](#)
- deleting user-defined [1192](#)
- DuplicateFileData ACE [1171](#)
- editing user-defined [1191](#)
- KeyPath ACE [1173](#)
- metrics [1192](#)
- MoveFileData ACE [1174](#)
- overview [1141](#)
- Predeployment Test [1601](#)
- reference [1142](#)

- RemoveFileData ACE [1176](#)
- RemoveIniFileData ACE [1176](#)
- RemoveRegistryData ACE [1177](#)
- specifying the Visual Studio C++ type library file path [1189](#)
- user-defined [1183](#)
- viewing ACE metrics [1192](#)
- WTS01 [1179](#)
- WTS02 [1179](#)
- WTS03 [1180](#)
- WTS04 [1181](#)
- WTS05 [1182](#)
- Action Element [681](#)
- activation [73](#)
 - ports used in [73](#)
 - silent [73](#)
- Active Directory [183](#)
 - Active Directory Configurations Page [183](#)
 - automatic login to Workflow Manager [176](#)
 - controlling access to ThinApp applications [860](#)
 - importing users [179](#)
 - LDAP attributes [183](#)
- Active Directory Configurations Page [183](#)
- Active Directory User Import View [179](#)
- Add AppLink Reference dialog box [886](#), [918](#)
 - entering relative paths [919](#)
- Add Ignore Table dialog [1231](#)
- Add New Tool dialog [144](#), [343](#)
- Add Tool Wizard [145](#), [152](#), [153](#)
 - Command-Line Configurations panel [153](#)
 - Command-Line Properties dialog [145](#)
 - Tool Properties panel [152](#)
 - Welcome panel [152](#)
- Add/Remove Program settings [1049](#)
 - changing [1049](#)
- Add/Remove Programs [1050](#)
 - disabling Modify button in [1050](#)
 - disabling Remove button in [1050](#)
 - disabling Repair button in [1050](#)
- Add/Remove Programs view [1049](#), [1086](#)
 - changing properties in [1049](#)
- adding
 - ad-hoc test results [1290](#)
 - test result message [1292](#)
- Additional Information panel [1254](#), [1256](#)
 - Rules Wizard [1254](#), [1256](#)
- Additional Information Tab [1229](#), [1230](#)
- additional server locations [1047](#)
 - configuring [1047](#)
- Additional Setup Programs dialog box [463](#), [513](#)
 - Repackaging Wizard [463](#), [513](#)
- ad-hoc import [230](#)
- ad-hoc test results
 - about [1290](#)
 - adding [1290](#)
- administration
 - copying roles [198](#)
 - creating new roles [197](#)
 - deleting roles [198](#)
- Administrative Install panel [1465](#), [1477](#)
 - Distribution Wizard [1465](#), [1477](#)
- administrative installation [1436](#)
 - creating with Distribution Wizard [1436](#)
- AdminStudio [111](#), [112](#), [114](#), [115](#), [132](#), [133](#), [139](#), [140](#), [152](#)
 - activation [73](#)
 - Add Tool Wizard [152](#)
 - adding tools [152](#)
 - Application Catalog Replication tables [404](#)
 - Application Catalog table names [391](#)
 - Assistant Tab [136](#)
 - checking for updates [115](#)
 - configuring the interface [112](#)
 - conflict persistence tables [408](#)
 - ConflictSolver tables [391](#)
 - creating new Application Catalog [213](#)
 - Customer Experience Improvement Program (CEIP) [148](#)
 - database schema tables [386](#)
 - database server permissions [389](#)
 - database subscription tables [408](#)
 - default Application Catalog [216](#)
 - disconnecting from Application Catalog [220](#)
 - extended attributes tables [408](#)
 - Groups table [408](#)
 - Interface reference [133](#)
 - Job Manager tables [420](#)
 - launching applications [112](#)
 - menus [140](#)
 - minimum permissions [389](#)
 - minimum privileges [389](#)
 - MSI table mapping [391](#)
 - MSI tables [392](#)
 - MSM tables [396](#)
 - OS Security Patch Wizard Tables [425](#)
 - OS Snapshot tables [401](#)
 - package and transform file tables [408](#)
 - Process Assistant [82](#), [84](#), [85](#), [86](#), [87](#), [88](#), [89](#), [90](#), [91](#), [92](#), [93](#), [94](#), [99](#), [100](#), [101](#), [102](#), [103](#), [104](#), [105](#), [106](#), [107](#), [108](#), [109](#), [110](#), [444](#)
 - Projects tab [139](#)
 - Quick Start evaluation experience [210](#)
 - sample database [210](#)
 - Security tables [402](#)

- setting shared location [114](#)
- setting task page help location [115](#)
- specifying default Application Catalog [216](#)
- specifying shared Application Catalog location [114](#)
- specifying the location of the AdminStudio Shared Directory [114](#)
- SQL Express [210](#)
- Start Page [134](#)
- System tables [391](#)
- Tool Properties dialog [149](#)
- toolbar [140](#)
- Tools Gallery [137](#)
- Tools tab [137](#)
- upgrading legacy databases [221](#)
- using [1011](#)
- using the Interface [111](#)
- virtualization [644](#)
- Workflow Manager tables [402](#)
- Workflow tables [401](#)
- AdminStudio Application Catalog [213](#), [216](#), [220](#), [260](#), [391](#)
 - creating new [213](#)
 - disconnecting from [220](#)
 - specifying default [216](#)
 - table names [391](#)
- AdminStudio databases [221](#)
 - upgrading [221](#)
- AdminStudio Enterprise Server
 - evaluating [73](#)
 - forgetting your password [172](#)
 - Report Center [1695](#)
 - standalone Application Catalogs [209](#)
- AdminStudio Interface [111](#), [112](#), [116](#), [133](#), [138](#), [143](#), [152](#)
 - configuring the interface [112](#)
 - configuring to stay on top [116](#)
 - dialogs [143](#)
 - reference [133](#)
 - using [111](#)
 - wizards [152](#)
 - Workflows Templates tab [138](#)
- AdminStudio options [145](#)
- AdminStudio Shared Directory [114](#), [145](#)
 - configuring for Replication [277](#)
 - Global Exclusions list [114](#)
- AdminStudio tasks [120](#)
 - associating tools with [120](#)
- AdminStudio tools [120](#), [121](#), [137](#)
 - deleting command-line configurations from [120](#)
 - deleting tools from Tools Gallery [121](#)
- AdminStudio Workflow Manager. See Workflow Manager
- Advanced Client [1499](#)
- advanced conversion options [541](#)
 - configuring Repackager [541](#)
- Advanced Marimba Options dialog box [1471](#)
- Advanced Options dialog [372](#), [1365](#), [1366](#), [1367](#)
 - Digital Signature tab [1367](#)
 - Manifest Options tab [1366](#)
- Advanced Options view [541](#)
- Advanced Settings view [612](#)
- advertisement [1024](#)
 - feature [1024](#)
- advertisements [1499](#), [1519](#)
- Advertisements View [1541](#)
- All Merge Modules view [338](#)
- All Reports page [1722](#)
- Allow application execution to the following user groups [860](#)
- ALLUSERS [628](#)
- alternate-language repackaging on clean machines [450](#)
- Altiris [1437](#)
 - Altiris Integration Panel [1465](#)
 - distribution [1437](#)
 - XML template [1466](#)
- ams_ [387](#)
- AMS. See Workflow Manager
- amssp_ [388](#)
- analysis options [303](#)
 - configuring in OS Snapshot Wizard [303](#)
- Analysis Options dialog [379](#)
- Analysis Options dialog box [515](#)
 - Repackaging Wizard [515](#)
- anonymous
 - Customer Experience Improvement Program (CEIP) [148](#)
- anonymous account [173](#), [174](#)
 - logging in [174](#)
 - setting up [173](#)
- anti-virus software [572](#)
 - excluding directories when using Snapshot method [451](#)
- app [516](#)
- application
 - saving after being wrapped [1661](#)
 - testing a wrapped application [1662](#)
 - using after being wrapped by FlexWrap [1616](#)
- Application Catalog [114](#), [213](#), [216](#), [219](#), [220](#), [260](#), [391](#), [437](#), [1103](#), [1597](#)
 - AdminStudio default [216](#)
 - connecting to [211](#), [212](#)
 - connecting to a specific using command-line options [437](#)
 - connecting to existing [210](#)
 - connecting to existing in ConflictSolver [1103](#)
 - connecting to on the Predeployment Test Reports Web site [1597](#)
 - creating new [213](#)
 - disconnecting from [220](#)

- enabling Software Repository [251](#)
- integration with InstallShield Editor [1004](#)
- replication [276](#), [277](#), [281](#)
- sample [210](#)
- searching [219](#)
- specifying a default for an AdminStudio enterprise [216](#)
- specifying default [216](#)
- specifying shared location [114](#)
- standalone [209](#)
- table names [391](#)
- upgrading [221](#)
- upgrading 5.0 or 5.5 or 6.0 Application Catalogs [221](#)
- upgrading legacy [221](#)
- upgrading pre-AdminStudio 5.0 Application Catalogs [221](#), [386](#)
- using the Software Repository [250](#)
- version management [256](#)
- Application Catalog Replication [276](#)
 - command line options [426](#)
 - creating [286](#)
 - creating a new publication [286](#)
 - creating a new subscription [292](#)
 - deleting a publication [290](#)
 - dialogs [340](#)
 - editing a publication [289](#)
 - editing a publication access list [290](#)
 - managing publications [285](#)
 - managing subscriptions [291](#)
 - Publication Manager [285](#)
 - publishing [277](#), [281](#), [284](#)
 - publishing a publication [288](#)
 - publishing to [276](#)
 - reference [304](#)
 - setting a publication schedule [289](#)
 - subscribing [277](#), [281](#), [291](#)
 - subscribing to [276](#)
 - subscribing to in Application Manager [266](#)
 - Subscription Manager [291](#)
 - tables [404](#)
 - tables excluded from replication [404](#)
 - tables included in replication [405](#)
 - tables included/excluded in [404](#)
 - wizards [361](#)
- Application Catalogs
 - merge replication using SQL Server [296](#)
- Application Configuration view [1085](#)
 - Tuner [1085](#)
- Application Conflict Evaluators [1141](#), [1142](#), [1146](#), [1147](#), [1148](#), [1149](#), [1151](#), [1152](#), [1153](#), [1155](#), [1157](#), [1158](#), [1159](#), [1160](#), [1161](#), [1162](#), [1163](#), [1164](#), [1165](#), [1167](#), [1168](#), [1183](#), [1191](#), [1192](#)

- ACE02 [1146](#)
- ACE03 [1146](#)
- ACE04 [1147](#)
- ACE05 [1148](#)
- ACE06 [1149](#)
- ACE07 [1149](#)
- ACE08 [1151](#)
- ACE09 [1152](#)
- ACE10 [1153](#)
- ACE12 [1155](#)
- ACE13 [1157](#)
- ACE14 [1158](#)
- ACE15 [1159](#)
- ACE16 [1160](#)
- ACE17 [1161](#)
- ACE19 [1162](#), [1163](#)
- ACE20 [1163](#)
- ACE21 [1164](#)
- ACE22 [1165](#)
- ACE23 [1167](#)
- ACE24 [1168](#)
- custom [1183](#)
- deleting user-defined [1192](#)
- DuplicateFileData ACE [1171](#)
- editing user-defined [1191](#)
- KeyPath ACE [1173](#)
- metrics [1192](#)
- MoveFileData ACE [1174](#)
- overview [1141](#)
- reference [1142](#)
- RemoveFileData ACE [1176](#)
- RemoveIniFileData ACE [1176](#)
- RemoveRegistryData ACE [1177](#)
- user-defined [1183](#), [1192](#)
- application isolation [552](#), [556](#), [1353](#), [1356](#), [1357](#), [1358](#), [1359](#), [1360](#), [1361](#), [1362](#)
- assemblies [1356](#)
- certificate store [1359](#)
- certificates [1358](#)
- code signing [1358](#)
- concept [1353](#)
- digital signatures [1358](#)
- filtering file listings when manually configuring [1362](#)
- manifests [552](#), [1357](#)
- methods [1356](#)
- modifying default recommendations [1361](#)
- private key [1359](#)
- setting assembly naming conventions [1360](#)
- software publishing credentials [1359](#)
- using Repackager vs. Application Isolation Wizard [1355](#)
- Application Isolation Wizard [1353](#), [1355](#), [1356](#), [1358](#), [1359](#),

- 1363, 1364, 1365, 1368, 1369, 1370
- Advanced Options dialog 1365
- Application Isolation Progress panel 1365
- Application Manifest Properties dialog 1369
- Assembly Properties dialog 1368
- AssemblyType property 1370
- CertificateFile property 1370
- CertificateName property 1370
- code signing technologies 1358
- command-line options 1370
- Company property 1370
- comparison to isolation using Repackager 1355
- Completing the Application Isolation Wizard panel 1365
- configuration files 1370
- digital signatures 1358, 1367
- Division property 1370
- IsolatedComponents property 1370
- Isolation Method panel 1364
- isolation methods 1356
- launching 1355
- Manifest and Assembly Design dialog 1368
- Manifests property 1370
- NewComponents property 1370
- private key 1359
- PVKFile property 1370
- reference 1363
- setting manifest options 1366
- software publishing credentials 1359
- SPCFile property 1370
- Summary panel 1364
- TimeStamp property 1370
- using Repackager to isolate Windows Installer packages 1355
- Welcome panel 1363
- Windows Installer File Selection panel 1364
- Application Link 885
- Application Manager 203, 222
 - adding groups 222
 - All Merge Modules view 338
 - Application Catalog database 405, 408
 - applying patches during command-line import 435
 - applying transforms during command-line import 435
 - checking out packages 257
 - command line 426
 - Components view 339
 - connecting to a specific Application Catalog using
 - command-line options 437
 - context menus 312
 - copying packages to multiple groups 223
 - creating a shortcut to a specific Application Catalog 438
 - deleting groups 223
 - deleting packages 230
 - Dependency view 339
 - editing groups 223
 - Exclusion view 339
 - extended attribute description file 225
 - extended attributes 224
 - extended attributes and Workflow Manager 227
 - Extended Attributes view 321, 326, 329, 330
 - Files view 340
 - Files/Components view 322
 - getting a copy of package in Software Repository 258
 - Group view 319
 - identifying packages in Software Repository 255
 - Import Wizard 364, 365, 373
 - importing 230, 234, 238, 245, 246, 248, 250
 - importing all packages in a directory using command line 436
 - importing Marimba NCP files into 247
 - importing merge modules 244
 - importing merge modules and Windows Installer
 - packages simultaneously 436
 - importing MSI packages into 232
 - importing OS snapshots into 245
 - importing other setup types into 249
 - importing packages in multiple configurations into 234
 - importing Windows Installer packages into 232
 - INI File Changes view 323, 327, 330
 - linking to a network directory 272
 - linking to a remote Application Catalog 265, 266, 269
 - Marimba NCP Files 329
 - menus 305
 - Merge Module view 326, 338
 - Merge Modules View 311
 - Merge Modules view 324
 - merging Application Catalog information 264
 - modifying groups 223
 - monitoring a Network Directory 265
 - moving groups in 224
 - moving OS snapshots in 224
 - moving products in 224
 - organizing products into groups 222
 - OS Snapshot view 326
 - OSD Files 329
 - Other Setup Types view 328
 - Output Window 316
 - Package Auto Import 266, 271, 272
 - Package Auto Import Wizard 373, 374, 375, 376
 - patches 334
 - Patches Group View 334
 - Patches tab 334, 336
 - Patches view 334, 336

- Product view [308](#), [319](#), [340](#)
- refreshing a linked package [270](#)
- Registry view [323](#), [327](#), [331](#)
- removing groups [223](#)
- running import silently [437](#)
- searching the Application Catalog [219](#)
- Shortcuts view [324](#), [328](#), [332](#)
- subscribing to an Application Catalog [266](#)
- supporting tables [408](#)
- system tables [405](#)
- Tables view [325](#), [328](#), [332](#)
- toolbar [305](#)
- user permissions [317](#)
- using a configuration file during command-line import [434](#)
- using a configuration file to import multiple merge modules [436](#)
- using a configuration file to import multiple Windows Installer packages [435](#)
- using extended attributes [225](#)
- viewing linked packages in Product View [271](#)
- Application Manager dialogs
 - Advanced Options [372](#)
 - Command-Line Parameters [343](#)
 - Extended Attribute Property [350](#)
 - Find [348](#)
 - Group Properties [350](#)
 - Package Auto Import Properties [352](#), [354](#)
 - Subscriptions [351](#)
- Application Manager extended attributes [225](#)
 - using [225](#)
- Application Manager groups [222](#), [223](#)
 - adding [222](#)
 - deleting [223](#)
 - editing [223](#)
 - modifying [223](#)
 - removing [223](#)
- Application Manager Output Window [316](#)
- Application Manager views
 - All Merge Modules [338](#)
 - Components [339](#)
 - Dependency [339](#)
 - Exclusion [339](#)
 - Files [340](#)
 - Files/Components [322](#)
 - Group [319](#)
 - INI File Changes [323](#), [327](#), [330](#)
 - Merge Module [326](#), [338](#)
 - Merge Modules [324](#)
 - OS Snapshot [326](#)
 - Other Setup Types [328](#)
 - Product [319](#)
 - Products [340](#)
 - Registry [323](#), [327](#), [331](#)
 - Shortcuts [324](#), [328](#), [332](#)
 - Tables [325](#), [328](#), [332](#)
- Application Manifest Properties dialog [1369](#)
- application manifests [552](#), [1357](#)
- application object template files [528](#)
- Application Sync [888](#)
- application virtualization [646](#)
- applications [1359](#)
 - isolating [1359](#)
 - version management [256](#)
- Applications page [829](#), [901](#)
 - creating a new application shortcuts [804](#), [875](#)
 - excluding or deleting a shortcut [805](#), [876](#)
 - including an existing shortcut [805](#), [876](#)
- AppLink [885](#)
 - adding reference [886](#)
 - Required and optional links [887](#)
- AppLink Reference dialog box
 - relative links [887](#)
- AppLink references [916](#)
- AppLink settings [915](#)
 - collisions and order of import [917](#)
 - entering a relative path [919](#)
 - example references [920](#)
 - required and optional linked applications [917](#)
 - required vs. optional [920](#)
 - Required vs. Optional applications [917](#)
- AppLink Settings dialog box [886](#), [915](#)
 - deleting a reference [887](#)
- AppSync Settings
 - Expiration tab [923](#)
- AppSync settings [888](#)
 - benefits of [888](#)
 - Clear Sandbox option [889](#)
 - configuring [888](#)
 - expiration [921](#)
 - Expire Period [923](#)
 - Expire Period option [890](#)
 - Frequency of update [889](#)
 - frequency of update [889](#)
 - Message displayed [889](#)
 - messages [890](#)
 - setting expiration settings [889](#)
 - URL of update [889](#)
 - Use Application Expiration [923](#)
 - Warning Frequency [924](#)
 - Warning Frequency option [890](#)
 - Warning Message [924](#)

- Warning Period [924](#)
- Warning Period option [890](#)
- AppSync Settings dialog box [921](#)
- App-V 4.6 [655](#)
- App-V Application Launcher [654](#), [691](#)
 - controlling whether to include in build output [693](#)
 - location of in build output [691](#)
 - opening [692](#)
 - requirements for using [692](#)
- App-V applications [516](#), [609](#), [643](#), [645](#), [695](#)
 - about [651](#), [773](#), [776](#)
 - about virtual suites [649](#)
 - adding diagnostic tools to [790](#), [841](#)
 - adding existing folder [794](#)
 - adding files to [792](#)
 - adding or deleting registry keys and values [808](#)
- App-V Assistant [771](#)
- App-VPackage directory [657](#)
- automatically creating from Repackager [646](#)
- benefits of using App-V Assistant to create [774](#)
- benefits of using Windows Installer Converter to create [651](#)
- building [816](#)
- building a Windows Installer package with build output [814](#)
- building using command line [849](#)
- command line [516](#)
- components of [656](#), [776](#)
- compressing packages for [813](#)
- conversion error and warning messages [850](#)
- creating 64-bit App-V applications [655](#)
- creating using InstallShield Microsoft App-V Assistant [645](#)
- creating with InstallShield [771](#)
- defining shortcut executables [804](#)
- deployment server [824](#)
- deployment server protocols [824](#)
- editing OSD file [658](#)
- editing OSD file to identify location of App-V server [657](#)
- evaluating AdminStudio's App-V support [75](#)
- excluding vs. deleting a shortcut [807](#)
- feature block 1 [846](#)
- feature block 2 [846](#)
- feature blocks [814](#)
- files included in [776](#)
- how transforms are included [781](#)
- ICO file [657](#)
- including additional Windows Installer packages [813](#)
- including an existing shortcut [805](#)
- including diagnostic tools [823](#)
- infrastructure of Microsoft Application Virtualization [773](#)
- inheritance of isolation options from folders to files [802](#)
- inheritance of isolation options in the registry [809](#)
- integration of Windows Services [658](#)
- launching [776](#)
- location of generated files [545](#), [657](#), [778](#)
- managing files and folders [792](#)
- Manifest file [657](#)
- methods to convert Windows Installer packages [644](#)
- modifying registry [807](#), [808](#)
- moving files and folders [796](#)
- OSD file [657](#)
- overview of [772](#)
- overview of isolation options [800](#)
- package optimizations [814](#)
- Package.DAT file [776](#)
- renaming a shortcut [807](#)
- selecting application shortcuts [803](#)
- setting file isolation options [843](#)
- setting folder isolation options [844](#)
- setting isolation options [800](#)
- setting isolation options for folders and files [801](#)
- SFT file [656](#)
- shortcut requirements [804](#), [829](#)
- specifying deployment server [823](#)
- specifying host name [824](#)
- specifying operating system requirements [823](#)
- specifying package name [787](#), [822](#)
- specifying port [824](#)
- specifying root folder name [822](#)
- specifying root folder name of SFT package [787](#)
- specifying virtual application server [824](#)
- start up and shut down sequences [659](#)
- steps to create [786](#)
- steps to create with App-V Assistant [779](#)
- storage path of application [825](#)
- support for Windows services [658](#), [782](#)
- testing using App-V Application Launcher [691](#)
- virtualization [695](#)
- App-V Assistant [544](#), [645](#), [650](#), [771](#), [826](#), [849](#)
 - about [778](#)
 - adding an existing folder to an App-V application [794](#)
 - adding diagnostic tools to an App-V application [790](#), [841](#)
 - adding files to an App-V application [792](#)
 - adding or deleting registry keys and values from an App-V application [808](#)
 - application features requiring pre- or post-conversion actions [850](#)
 - Applications page [829](#)
 - benefits of using [774](#)
 - Build Options page [835](#)
 - building a Windows Installer package with build output

- 814
- building an App-V application 816
- building App-V application in Direct Edit mode 812
- compared with Microsoft App-V Sequencer 775
- controlling the display of predefined folders 796
- creating 64-bit App-V applications 655
- creating an App-V application 786
- creating new application shortcut executables 804
- creating new folder 795
- deleting files and folders 796
- deployment server 789
- deployment server host 789
- deployment server port 790
- Diagnostic Tools dialog box 790, 841
- Dynamic Suite Composition page 832
- enabling App-V application building when editing a Windows Installer package 812
- error messages 850
- excluding or deleting an App-V application shortcut 805
- FILE protocol 789
- Files page 826
- Home page 820
- how transforms are included 781
- HTTP protocol 789, 824
- HTTPS protocol 789
- including an existing shortcut 805
- inheritance of isolation options 802
- inheritance of isolation options in registry 809
- integration with Project Assistant and Installation Designer 768
- interface comparison to Microsoft App-V Sequencer 783
- Launch App-V Application dialog box 845
- managing files and folders 792
- modifying build options 811
- modifying registry settings 807
- modifying shortcuts 803
- moving files and folders 796
- overview of 772
- overview of isolation options 800
- Package Information page 821
- Package Optimizations dialog box 815, 845
- performing dynamic suite composition 810
- Primary Application Directory dialog box 847
- reference 819
- Registry Isolation Options dialog box 848
- Registry page 831
- renaming a shortcut 807
- RTSP protocol 789
- RTSPS protocol 789
- selecting releases to build 812
- setting isolation options 800
- setting isolation options for folders and files 801
- setting isolation options on files 843
- setting isolation options on folders 844
- specifying deployment server 789
- specifying general settings 786
- specifying OS requirements 788
- specifying primary application directory 798
- specifying upgrade information 788
- steps to create an App-V application 779
- support for Windows services 782
- supported InstallShield project types 781
- when to exclude and when to delete shortcuts 807
- App-V Client 651
- App-V infrastructure 651
- App-V Package Upgrade Settings dialog box 842
- App-V Sequencer 651
 - clean machine requirement 653
 - installation directory requirement 653
 - vs. Windows Installer Converter 651
- App-V Server 651
- App-V server location
 - in OSD file 657
- ARPCOMMENTS 1086
- ARPCONTACT 1086
- ARPHHELPINK 1086
- ARPHELPTELEPHONE 1086
- ARPNOMODIFY 1086
- ARPNOREMOVE 1086
- ARPNOREPAIR 1086
- ARPURLINFOABOUT 1086
- ARPURLUPDATEINFO 1086
- as 388
- as-hoc import 236, 250
- AsJob 420
- AsJobQueue 421
- AsJobQueueStepDetail 423
- AsJobStep 422
- AsJobStepMaster 423
- ASP.NET permissions 1597
 - assigning in Predeployment Test 1597
- AsQueueStatus 424
- AsSchedule 425
- assemblies 556, 1356, 1358, 1359, 1360, 1363
 - digital signatures 1358
 - in application isolation 1356
 - private key 1359
 - servicing published shared 1363
 - setting naming convention 556, 1360
 - signing 1358
 - software publishing credentials 1359
- assemblies and manifests 1356

- isolation method [1356](#)
- assembly files [550](#)
- assembly manifests [552](#), [1357](#)
- Assembly Properties dialog [1368](#)
- AssemblyType [1370](#)
- Assistant Tab [136](#)
- Associate with Workflow Manager Application dialog [342](#)
- Associated Patches view [1219](#)
- associated tools [121](#), [128](#)
 - running in projects [121](#), [128](#)
- assys [387](#)
- attributes [183](#)
 - creating Active Directory [183](#)
- authentication modes [268](#)
 - preferred for replication [268](#)
- authoring packages [1001](#)
- automatic job execution
 - Job Manager [1753](#)
- automatic login [176](#)
 - using Active Directory [176](#)
- automation
 - Job Manager [1753](#)

B

- b [516](#)
- Batch Conversion conversion type [650](#)
- Batch Convert option [700](#)
- batch convert option
 - evaluating [75](#)
- best practice ACEs [95](#)
- best practices [302](#)
 - ACEs [95](#)
 - ConflictSolver [1104](#), [1106](#), [1119](#), [1122](#)
 - OS Snapshot [302](#)
- best practices ACEs [1171](#), [1174](#), [1176](#), [1177](#)
- Best Practices Page [95](#)
- Build App-V application [836](#)
- Build Options page [835](#), [904](#)
- Build Settings page [987](#)
 - digitally signing a Citrix profile [971](#)
- Build ThinApp application [907](#)
- Build Virtual Package [836](#), [907](#)
- BuildCompressed [628](#)
- building App-V applications using command line [849](#)
- building profiles using command line [999](#)
- building ThinApp applications using command line [925](#)
- business rules [1182](#), [1184](#)

C

- C [426](#), [434](#), [435](#), [436](#)
- C1083 error [1189](#)
- CAB file [665](#)
- CAB files [1027](#)
 - Tuner [1027](#)
- CAPS (Client Access Points) [1499](#)
- Captured Installation view [596](#)
- capturing OS snapshots [302](#), [304](#)
- CARD [1141](#), [1146](#), [1147](#), [1148](#), [1149](#), [1159](#), [1162](#), [1163](#), [1194](#), [1196](#), [1197](#), [1198](#), [1199](#)
- CARD02 [1146](#), [1196](#)
- CARD04 [1147](#), [1196](#)
- CARD05 [1148](#), [1197](#)
- CARD06 [1149](#), [1197](#)
- CARD07 [1149](#), [1197](#)
- CARD15 [1159](#), [1198](#)
- CARD18 [1162](#), [1198](#)
- CARD19 [1163](#), [1198](#)
- CARD20 [1163](#), [1199](#)
- CARDs [1195](#)
 - CARD02 [1196](#)
 - CARD04 [1196](#)
 - CARD05 [1197](#)
 - CARD06 [1197](#)
 - CARD07 [1197](#)
 - CARD15 [1198](#)
 - CARD18 [1198](#)
 - CARD19 [1198](#)
 - CARD20 [1199](#)
 - overview [1194](#)
 - reference [1195](#)
- CEIP (Customer Experience Improvement Program) [148](#)
- certificate name in the store [1358](#)
- CertificateFile [1370](#)
- CertificateName [1370](#)
- certificates [1358](#), [1359](#)
 - certificate store [1359](#)
 - private key [1359](#)
- cf [516](#)
- channel publishing [1441](#)
 - using Marimba with Distribution Wizard [1441](#)
- check in and check out functionality of packages [257](#)
- checking for updates [145](#)
- checklist [1062](#)
- Choose Registry Key dialog box [627](#)
- Citrix [516](#), [543](#), [610](#), [662](#), [668](#), [695](#)
 - Citrix profile [532](#), [610](#)
 - Citrix XenApp [543](#), [662](#), [668](#)
 - command line [516](#)

- virtualization [695](#)
- Windows Installer Converter [695](#)
- Citrix Assistant [544](#), [645](#), [935](#), [999](#)
 - about [938](#)
 - adding an existing folder to a profile [954](#)
 - adding diagnostic tools to profile [945](#), [991](#)
 - adding files to a Citrix profile [953](#)
 - adding or deleting registry keys and values from a Citrix profile [968](#)
 - adding pre-launch and post-exit scripts to profile [950](#)
 - application features requiring pre- or post-conversion actions [999](#)
 - Build Settings page [987](#)
 - building a Citrix profile [972](#)
 - building Citrix profile in Direct Edit mode [972](#)
 - controlling the display of predefined folders [956](#)
 - creating a Citrix profile [943](#)
 - creating new folder [955](#)
 - creating new profile shortcut [964](#)
 - deleting files and folders [956](#)
 - Diagnostic Tools dialog box [945](#), [991](#)
 - digitally signing a profile [971](#)
 - enabling a profile to run non-included executables [945](#)
 - enabling Citrix profile building when editing a Windows Installer package [972](#)
 - error messages [999](#)
 - excluding or deleting a profile shortcut [965](#)
 - Home page [975](#)
 - how requirements are applied at runtime [949](#)
 - how transforms are included [942](#)
 - including an existing profile shortcut [964](#)
 - inheritance of isolation options [960](#)
 - inheritance of isolation options in registry [969](#)
 - jumping to from Windows Installer Converter [705](#)
 - managing files and folders [952](#)
 - modifying build settings [969](#)
 - modifying registry settings [967](#)
 - modifying shortcut settings [961](#)
 - moving files and folders [956](#)
 - opening from Windows Installer Converter [668](#)
 - overview [935](#)
 - overview of [935](#)
 - overview of Citrix profiles [660](#), [939](#)
 - overview of isolation options [958](#)
 - Profile Files page [981](#)
 - Profile Information page [976](#)
 - Profile Registry page [986](#)
 - Profile Requirements page [978](#)
 - Profile Shortcuts page [984](#)
 - reference [974](#)
 - Registry Isolation Options dialog box [996](#)

- renaming a shortcut [966](#)
- Script Execution dialog box [991](#)
- selecting releases to build [970](#)
- Service Packs Requirement dialog box [998](#)
- setting isolation options [957](#)
- setting isolation options for folders and files [960](#)
- setting isolation options on files [993](#)
- setting isolation options on folders [994](#)
- setting language requirements [948](#)
- setting operating system and service pack requirements [947](#)
- setting registry isolation options [968](#)
- setting Service Pack requirements [998](#)
- specifying profile information [943](#)
- specifying profile name and version [944](#)
- specifying profile requirements [947](#)
- steps to create a Citrix profile [936](#)
- supported InstallShield project types [781](#), [942](#)
- when to exclude or delete shortcuts [966](#)
- Citrix Presentation Server 4.5
 - benefits of [852](#)
- Citrix profile [532](#), [543](#), [610](#), [643](#), [645](#), [662](#), [663](#), [668](#), [695](#)
 - about custom actions [761](#)
 - about services [761](#)
 - and COM+ applications [763](#)
 - automatically creating from Repackager [646](#)
 - benefits of deploying [666](#)
 - building using Windows Installer Converter [669](#)
 - CAB file [665](#)
 - creating using InstallShield Citrix Assistant [645](#)
 - creating using Repackager [668](#)
 - creating using Windows Installer Converter [668](#)
 - features requiring pre- or post-conversion actions [761](#)
 - hash file key for digital signatures [665](#)
 - hashes file [665](#)
 - ignored tables [761](#)
 - location of generated files [664](#)
 - methods to convert Windows Installer packages [644](#)
 - overview of [660](#), [664](#)
 - overview of Citrix XenApp [663](#)
 - profile manifest file [665](#)
 - testing prior to conversion [673](#)
- Citrix profiles [937](#)
 - about [939](#)
 - about virtual suites [649](#)
 - adding diagnostic tools to [945](#), [991](#)
 - adding existing folder [954](#)
 - adding files to [953](#)
 - adding or deleting registry keys and values [968](#)
 - adding pre-launch and post-exit scripts [950](#)
 - adding scripts [950](#)

- benefits of [852](#)
- benefits of deploying [940](#)
- building using command line [999](#)
- CAB file [939](#)
- Citrix Assistant [935](#)
- conditions when Shortcut should be excluded or deleted [966](#)
- conversion error and warning messages [999](#)
- creating with InstallShield [935](#)
- digitally signing [971](#)
- enabling a profile to run non-included executables [945](#)
- hashes.txt file [939](#)
- how requirements are applied at runtime [949](#)
- how shortcuts are implemented [962](#)
- how transforms are included [942](#)
- Icons file [940](#)
- icons.bin file [940](#)
- including an existing shortcut [964](#)
- inheritance of isolation options from folders to files [960](#)
- inheritance of isolation options in the registry [969](#)
- managing files and folders [952](#)
- moving files and folders [956](#)
- myapp.profile file [939](#)
- overview of [939](#)
- overview of Citrix Assistant [935](#)
- overview of Citrix XenApp [937](#)
- overview of isolation options [958](#)
- Profile Manifest file [939](#)
- renaming a shortcut [966](#)
- Scripts folder [940](#)
- setting file isolation options [993](#)
- setting folder isolation options [994](#)
- setting isolation options [957](#)
- setting isolation options for folders and files [960](#)
- setting language requirements [948](#)
- setting operating system and service pack requirements [947](#)
- setting registry isolation options [968](#)
- setting Service Pack requirements [998](#)
- shortcut requirements [963](#)
- shortcuts and the isolation environment [962](#)
- steps to create with Citrix Assistant [936](#)
- Citrix XenApp [662](#), [663](#), [935](#), [937](#)
 - about [937](#)
 - about COM+ applications [763](#)
 - and services [762](#)
 - benefits of [666](#)
 - overview of [663](#), [937](#)
 - overview of Citrix profiles [660](#), [664](#), [939](#)
- Citrix XenApp Assistant. See Citrix Assistant.
- class IDs [1384](#)
 - checking [1384](#)
- Class IDs view [1421](#)
- clean machines [449](#), [450](#)
 - alternate-language repackaging on [450](#)
 - Repackager [449](#)
- Clear Sandbox [889](#)
- Client Access Points (CAPS) [1499](#)
- Client Agent [1499](#)
- code signing [1358](#)
 - certificate name in the store [1358](#)
 - credentials [1358](#)
- Collect Product Information panel [504](#)
 - Repackaging Wizard [504](#)
- collection [1499](#)
- COM+ applications
 - and Citrix profile [763](#)
 - and Citrix XenApp [763](#)
- command line [426](#), [435](#), [437](#), [493](#), [849](#), [925](#), [999](#), [1405](#)
 - applying patches to packages during import [435](#)
 - applying transforms to packages during import [435](#)
 - build App-V application [516](#)
 - build Citrix profile [516](#)
 - build ThinApp application [516](#)
 - creating a log file during import [437](#)
 - for Windows Installer Converter [668](#)
 - in Application Manager [426](#)
 - Predeployment Test [1560](#)
 - running QualityMonitor [1405](#)
 - running Repackaging Wizard from [493](#)
- Command tab [1064](#), [1235](#)
- command-line configurations [119](#), [120](#), [150](#)
 - adding [119](#)
 - creating [150](#)
 - deleting [150](#)
 - deleting from existing tools in AdminStudio [120](#)
 - editing [150](#)
 - modifying for existing tools [119](#)
- Command-Line Configurations panel [153](#)
 - Add Tool Wizard [153](#)
- command-line import [434](#)
 - using a configuration file in Application Manager [434](#)
- command-line options [435](#), [436](#), [437](#), [516](#), [1047](#), [1370](#)
 - Application Isolation Wizard [1370](#)
 - connecting to a specific Application Catalog [437](#)
 - Repackager [516](#)
 - using a configuration file to import multiple merge modules in Application Manager [436](#)
 - using a configuration file to import multiple Windows Installer packages in Application Manager [435](#)
 - when to use Dialogs view instead [1047](#)
- Command-Line Parameters dialog [343](#)

- command-line properties [145](#)
 - Add Tool Wizard [145](#)
 - Command-Line Properties dialog [145](#)
 - DevLocation variable [145](#)
 - InstallLocation variable [145](#)
 - ProjectName variable [145](#)
 - SharedPoint variable [145](#)
 - SourcePackage variable [145](#)
 - TargetDir variable [145](#)
 - TargetFileName variable [145](#)
- Command-Line Properties dialog [145](#)
 - Add Tool Wizard [145](#)
- COMMapping [628](#)
- Comments [1086](#)
- comments [1051](#)
 - adding and editing setup property [1051](#)
- Company [1370](#)
- Completing the Application Isolation Wizard panel [1365](#)
- component conflicts [1118](#)
- component settings options [542](#), [613](#)
- Components view [339](#)
- compression of ThinApp applications [874](#)
- Compression Type [907](#)
- configuration file [434](#), [435](#), [436](#)
 - Application Manager [435](#)
 - importing multiple merge modules using [436](#)
 - importing multiple Windows Installer packages using [435](#)
 - using with Application Manager command-line import [434](#)
- configuration files [1370](#)
 - Application Isolation Wizard [1370](#)
- Configuration Manager [1498](#)
- Configuration Manager Server [1499](#)
 - saving a package on using Configuration Manager Web Console [1522](#)
- Configuration Manager Web Console [1497](#), [1499](#), [1502](#)
 - access accounts [1510](#)
 - advertisements [1499](#), [1519](#)
 - Advertisements View [1541](#)
 - and Microsoft Configuration Manager [1498](#)
 - Client Access Points (CAPS) [1499](#)
 - Client Agent [1499](#)
 - collection [1499](#)
 - configuring distribution settings [1503](#)
 - distributing a package to Configuration Manager Server [1522](#)
 - distributing packages [1502](#), [1504](#)
 - Distribution Points [1499](#)
 - distribution points [1511](#)
 - Distribution Points View [1533](#)
 - Distribution Settings page [1544](#)
 - getting started [1502](#)
 - Management Points [1499](#)
 - modifying distribution settings [1522](#)
 - Package Administration Page [1545](#)
 - package definition file [1527](#)
 - package settings [1506](#)
 - programs [1512](#)
 - reference [1525](#)
 - saving a package on Configuration Manager Server [1522](#)
 - Select a Configuration Manager Collection dialog box [1544](#)
 - setting package configuration options [1505](#)
 - setting package configuration options\>\ [1505](#)
 - target collection [1499](#)
 - viewing status of distributed package [1522](#)
- Configuration tab [150](#)
 - Tool Properties dialog [150](#)
- configuring command lines [153](#)
- conflict analysis
 - using multiple source packages [1119](#)
- Conflict Application Resolution Definitions (CARDs) [1141](#), [1146](#), [1147](#), [1148](#), [1149](#), [1159](#), [1162](#), [1163](#), [1194](#), [1195](#)
 - CARD02 [1196](#)
 - CARD04 [1196](#)
 - CARD05 [1197](#)
 - CARD06 [1197](#)
 - CARD07 [1197](#)
 - CARD15 [1198](#)
 - CARD18 [1198](#)
 - CARD19 [1198](#)
 - CARD20 [1199](#)
 - overview [1194](#)
 - reference [1195](#)
- conflict identification [1118](#), [1126](#)
 - running again [1126](#)
- Conflict Information dialog [1232](#)
- conflict options [1119](#), [1237](#)
 - changing ConflictSolver default [1119](#)
 - ConflictSolver [1237](#)
- Conflict Options panel [1253](#)
 - Conflict Wizard [1253](#)
- conflict persistence [1126](#)
 - deleting persisted data [1126](#)
- conflict persistence tables [408](#)
- conflict resolution [1118](#)
- Conflict Resolution dialog [1232](#)
- conflict types [1119](#)
 - changing default in ConflictSolver [1119](#)
- Conflict Wizard [1122](#), [1248](#), [1249](#), [1250](#), [1252](#), [1253](#)
 - Conflict Options panel [1253](#)
 - identifying conflicts [1122](#)

- MSI Source Information panel [1250](#)
- Source Package panel [1250](#)
- Source Type panel [1250](#)
- Summary panel [1253](#)
- Target Information panel [1252](#)
- Welcome panel [1249](#)
- conflicts [1118](#), [1122](#), [1124](#), [1125](#), [1126](#), [1127](#), [1129](#), [1130](#), [1131](#)
 - automatically resolving in ConflictSolver [1129](#)
 - checking for during import [1124](#)
 - component [1118](#)
 - conflict resolution process in ConflictSolver [1127](#)
 - deleting persisted [1126](#)
 - file extension [1118](#)
 - files [1118](#)
 - identifying with Conflict Wizard [1122](#)
 - INI file [1118](#)
 - manually resolving in ConflictSolver [1130](#)
 - NT services [1118](#)
 - ODBC resources [1118](#)
 - product code [1118](#)
 - registry [1118](#)
 - resolving [1127](#)
 - resolving directly in Windows Installer package [1131](#)
 - resolving using transforms [1131](#)
 - shortcut [1118](#)
 - solution using virtualization [647](#)
 - viewing identified [1125](#)
- Conflicts Tab [1208](#)
- Conflicts tab [1237](#)
 - in ConflictSolver Options dialog [1237](#)
- Conflicts view [1213](#), [1222](#)
- ConflictSolver [93](#), [1103](#), [1109](#), [1110](#), [1111](#), [1116](#), [1117](#), [1119](#), [1121](#), [1122](#), [1124](#), [1126](#), [1127](#), [1129](#), [1130](#), [1131](#), [1132](#), [1135](#), [1136](#), [1183](#), [1191](#), [1192](#), [1200](#), [1203](#), [1206](#), [1211](#), [1213](#), [1217](#), [1218](#), [1219](#), [1220](#), [1222](#), [1227](#), [1248](#), [1249](#), [1250](#), [1252](#), [1253](#), [1254](#), [1256](#), [1257](#), [1258](#), [1259](#)
 - ACE reference in Predeployment Test [1601](#)
 - All Patches View [1339](#)
 - Associated Patches View [1339](#)
 - Associated Patches view [1219](#)
 - automatically resolving conflicts [1129](#)
 - best practices [1104](#), [1106](#), [1119](#), [1122](#)
 - changing default conflict options [1119](#)
 - changing default conflict types [1119](#)
 - changing default validation options [1116](#)
 - changing resolution options [1121](#)
 - changing the default validation file in [1116](#)
 - checking for conflicts during import [1124](#)
 - conflict persistence [1126](#)
 - conflict resolution process [1127](#)
 - Conflict Wizard [1122](#), [1248](#), [1249](#), [1250](#), [1252](#), [1253](#)
 - Conflicts view [1213](#), [1222](#)
 - ConflictSolver database [1109](#)
 - ConflictSolver tab on InstallShield Editor Options dialog [1009](#)
 - connecting to an existing Application Catalog [1103](#)
 - Consolidated Patch Report [1343](#)
 - context menus [1206](#)
 - Crystal Reports [1135](#)
 - cssys tables [405](#)
 - cstbl tables [408](#)
 - custom [1183](#)
 - custom reports [1136](#)
 - deleting user-defined ACEs [1192](#)
 - Dependencies View [1339](#)
 - Dependencies view [1218](#)
 - dialogs [1227](#)
 - editing user-defined ACEs [1191](#)
 - handling invalid Windows Installer packages [1117](#)
 - ICEs and Validation Errors [1117](#)
 - Impact Analysis View [1339](#)
 - Impacts View on Patches Tab [1226](#)
 - manually resolving conflicts [1130](#)
 - Marimba NCP Files [1221](#)
 - menus [1200](#)
 - Merge Module view [1219](#)
 - OS Snapshot view [1219](#)
 - OSD Files [1221](#)
 - Other Setup Types view [1220](#)
 - Output Window [1208](#)
 - Patch Impact Analysis Wizard [1325](#), [1336](#), [1344](#)
 - Patch Impact Information dialog box [1246](#)
 - Patch Impact view [1217](#)
 - Patch Impacts View [1339](#)
 - patches [1222](#)
 - Patches Tab [1226](#)
 - Patches tab [1222](#), [1225](#), [1339](#)
 - Patches Tab Views [1222](#)
 - Patches view [1339](#)
 - performing patch impact analysis [1336](#)
 - persistence [1126](#)
 - Product View [1203](#), [1211](#)
 - reimporting packages [1132](#)
 - reports [1132](#), [1134](#)
 - resolution and the Software Repository [1758](#)
 - resolving conflicts [1127](#)
 - resolving conflicts directly in package [1131](#)
 - resolving conflicts using transforms [1131](#)
 - Rules Wizard [1253](#), [1254](#), [1256](#), [1257](#), [1258](#)
 - scanning packages for dependencies [1218](#)
 - tables [391](#)

- Tables view [1219](#), [1220](#)
- Test Results view [1217](#), [1269](#)
- Testing packages using PackageExpert [1217](#)
- testing packages using PackageExpert [1269](#)
- toolbar [1200](#)
- user-defined ACEs [1183](#)
- Validate Package function [1112](#)
- validating external Windows Installer packages [1110](#)
- validation [93](#), [1109](#), [1110](#)
- validation after import [1112](#), [1113](#), [1114](#), [1115](#)
- validation during import [1111](#)
- Validation Explanation dialog [1235](#)
- Validation view [1215](#)
- Validation Wizard [1258](#), [1259](#)
- viewing PackageExpert test results [1280](#)
- viewing patch and patch impact information [1339](#)
- viewing patch impact analysis results [1336](#), [1343](#)
- ConflictSolver Options dialog [1237](#), [1239](#), [1241](#), [1242](#), [1243](#), [1244](#)
 - Conflicts tab [1237](#)
 - Duplicate Package tab [1239](#)
 - Extended Attributes tab [1237](#), [1241](#)
 - General tab [1237](#), [1241](#)
 - Import tab [1237](#), [1242](#)
 - Resolution tab [1237](#), [1243](#)
 - Validate tab [1237](#), [1244](#)
- ConflictSolver Output Window [1208](#)
- Connect to Application Catalog dialog [343](#), [357](#)
- Connection to New Catalog Page (Predeployment Test) [1597](#)
- Consolidated Patch Report [1343](#)
- Contact Person [1086](#)
- context file [694](#)
- context menus [312](#), [1206](#)
 - Application Manager [312](#)
 - ConflictSolver [1206](#)
- contextual packages [1757](#)
- Control Access via Active Directory [860](#), [897](#)
- Conversion Mode panel [698](#)
 - Batch Convert option [700](#)
 - Single Application option [699](#)
 - Virtual Suite option [700](#)
- converting [526](#), [527](#), [531](#)
 - .aot [527](#)
 - .axt [527](#)
 - .inc [527](#)
 - .ipf [527](#)
 - .isl [531](#)
 - .txt [531](#)
 - .wse [531](#)
 - InstallScript MSI [492](#)
 - InstallShield log files [531](#)
 - legacy setups [526](#)
 - Novell ZENworks projects [527](#)
 - Repackager 3.x output [527](#)
 - SMS projects [527](#)
 - WinINSTALL projects [531](#)
 - Wise Installation projects [531](#)
- copying [223](#)
 - packages to multiple Application Manager groups [223](#)
 - roles [198](#)
- Create a new transform file option [1068](#)
- Create Element [681](#)
- Create Report dialog box [587](#)
- CreateSetupExe [628](#)
- creating
 - roles [197](#)
- creating new workflows [127](#)
- credentials [1358](#)
- Crystal Reports [1132](#)
 - generating in ConflictSolver [1135](#)
- cs [516](#)
- csmsi [387](#)
- csmsm [387](#)
- csosc [387](#)
- cspch [388](#)
 - OS Security Patch Wizard Tables [425](#)
- cssp_ [388](#)
- cssys [387](#)
- cssys ConflictSolver supporting tables [408](#)
- cssys ConflictSolver system tables [405](#)
- cssysExtendedAttributeSchema [405](#)
- cssysSchemaInfo [407](#)
- cssysTables [407](#)
- cstbl [387](#)
 - cstblConflictACE [411](#)
 - cstblConflictPackages [412](#)
 - cstblConflictResults [413](#)
 - cstblConflictResultsDestPkgs [414](#)
 - cstblExtendedAttributeData [419](#)
 - cstblGroupPackages [419](#)
 - cstblGroups [417](#)
 - cstblPackage [415](#)
 - cstblPackageHistoryLog [418](#)
 - cstblPackageRelationships [417](#)
 - cstblSubscribedData [409](#)
 - cstblSubscribedDatabases [409](#)
 - cstblTransformFile [416](#)
- custom actions
 - and Citrix profiles [761](#)
 - and Windows Installer Converter [761](#)
- custom installation [1023](#)

- preventing feature display [1023](#)
- custom rules [681](#)
- custom setups [1045](#)
 - disabling [1045](#)
- custom tables [1242](#)
- Custom Test Results Editor dialog box [1317](#)
- customer
 - Customer Experience Improvement Program (CEIP) [148](#)
- Customer Experience Improvement Program (CEIP) [148](#)
 - canceled membership [148](#)
 - joining [148](#)
- Customize dialog [1235](#)
- Customize dialog box [1064](#)

D

- D [426](#)
- data binding [690](#)
- data sources [1038](#)
 - adding new [1038](#)
- data type appearance [524](#)
 - changing Repackager [524](#)
- database
 - foreign key [388](#)
 - primary key [388](#)
 - sample [210](#)
- database schema tables [386](#)
 - prefixes [387](#)
- database subscription tables [408](#)
- debug log [116](#)
 - generating [116](#)
- Default Destination [1022](#)
- Default Organization [1022](#)
- Default.ini [635](#)
- Deleted Files view [604](#)
- Deleted Registry Entries view [605](#)
- Deleted.isr [635](#)
- deleting
 - filters in PackageExpert [1286](#)
 - groups [162](#)
 - roles [198](#)
 - test results in PackageExpert [1281](#)
 - users [162](#)
- deleting added setups [464](#)
- dependencies [547](#), [1108](#)
 - detecting in Repackager [547](#)
 - scanning for [1108](#)
 - viewing in ConflictSolver [1339](#)
- Dependencies view [1108](#), [1218](#)
- Dependency view [339](#)
- deploying SMS [1055](#)
- Deployment & Conflict Test Selection panel [1572](#)
- Deployment Status [1400](#)
 - install or configure products or features [1400](#)
 - reinstall components [1402](#)
 - reinstall features [1401](#)
 - verify files [1399](#)
 - view properties [1399](#)
- Deployment Status view [1431](#)
- deployment testing [1383](#)
 - Predeployment Test [1551](#), [1552](#)
- deployment tests [1420](#)
 - executing all from command line [1420](#)
 - executing all from Interface [1420](#)
 - running silently [1420](#)
- description file [225](#)
 - Application Manager extended attributes [225](#)
- Description property [1022](#)
 - editing for features [1022](#)
- description property [1023](#)
- Destination Group panel [373](#)
 - Import Wizard [373](#)
- destination variable [1023](#)
- DevLocation [145](#)
- diagnostic tools
 - adding to App-V application [790](#), [841](#)
 - adding to Citrix profile [945](#), [991](#)
 - adding to ThinApp application [861](#), [910](#)
- Diagnostic Tools dialog box [790](#), [841](#), [861](#), [910](#), [945](#), [991](#)
- dialog [1046](#)
 - editing properties for [1046](#)
- dialog boxes [1227](#)
 - ConflictSolver [1227](#)
- Dialog Properties dialog [1066](#)
- dialog sequences [1045](#)
 - restoring [1045](#)
- dialogs [143](#), [144](#), [145](#), [149](#), [1044](#), [1045](#), [1047](#), [1064](#), [1066](#), [1365](#)
 - About AdminStudio [143](#)
 - Add New Tool [144](#)
 - AdminStudio Interface [143](#)
 - AdminStudio Options [145](#)
 - Advanced Options [1365](#)
 - Command-Line Properties in Add Tool Wizard [145](#)
 - Dialog Properties dialog [1066](#)
 - hiding during UI sequences [1044](#)
 - suppressing License Agreement [1045](#)
 - suppression issues [1047](#)
 - Tool Properties [149](#)
 - Transform Summary [1064](#)
 - working with in Tuner [1044](#)
- Dialogs view [1044](#), [1045](#), [1046](#), [1047](#), [1066](#), [1086](#)

Index

- Dialog Properties dialog [1066](#)
- disabling custom setups from [1045](#)
- editing dialog properties in [1046](#)
- hiding dialogs during UI sequences from [1044](#)
- restoring dialog sequences from [1045](#)
- suppressing License Agreement dialog from [1045](#)
- when to use over MSI command-line options [1047](#)
- digital certificates [1358](#), [1359](#)
 - certificate store [1359](#)
- Digital Signature tab [1367](#)
 - Advanced Options dialog [1367](#)
- digital signatures [1358](#), [1359](#), [1367](#)
 - Application Isolation Wizard [1367](#)
 - private key [1359](#)
 - software publishing credentials [1359](#)
- digitally signing
 - a Citrix profile [971](#)
- Direct Editor [1056](#), [1057](#), [1090](#)
 - adding a new record [1057](#)
 - adding a new row [1057](#)
 - editing packages [1056](#)
 - finding and replacing in [1057](#)
 - from Validation tab [1057](#)
- directories [567](#)
 - exclusions [567](#)
- Directories and Files Excluded During Analysis dialog box [626](#)
- directories and subdirectories [559](#)
 - excluding [559](#)
- directory exclusions [567](#)
- Directory Service Attributes [163](#)
- Directory Services [197](#)
 - automatic login [172](#)
 - creating Directory Service Attributes [163](#)
 - creating new role with Directory Services [197](#)
 - Directory Service Attributes [163](#)
 - managing Directory Services configurations [163](#)
- Directory Services Page [180](#)
- Disable Log Monitor Tracing [907](#)
- Disable Modify Button [1086](#)
- Disable Remove Button [1086](#)
- Disable Repair Button [1086](#)
- disabling [1050](#)
 - Modify button in Add/Remove Programs [1050](#)
 - Remove button in Add/Remove Programs [1050](#)
 - Repair button in Add/Remove Programs [1050](#)
- disabling a Subscription [283](#)
- disabling a user account [161](#)
- disconnecting from Application Catalog [220](#)
- distributing packages [1502](#)
- distribution [1052](#), [1053](#), [1435](#)
 - copying to FTP server [1053](#)
 - copying to network location [1052](#)
 - preparing packages [1435](#)
- Distribution Output panel [1486](#)
 - Distribution Wizard [1486](#)
- Distribution Points [1499](#)
- distribution points [1511](#)
- distribution settings [1522](#)
 - modifying in Configuration Manager Web Console [1522](#)
- Distribution Summary panel [1486](#)
 - Distribution Wizard [1486](#)
- Distribution Type panel [1464](#)
 - Distribution Wizard [1464](#)
- Distribution Wizard [1435](#), [1436](#), [1437](#), [1439](#), [1441](#), [1443](#), [1444](#), [1445](#), [1446](#), [1452](#), [1456](#), [1461](#), [1463](#), [1464](#), [1465](#), [1467](#), [1468](#), [1469](#), [1470](#), [1471](#), [1472](#), [1473](#), [1474](#), [1477](#), [1480](#), [1482](#), [1483](#), [1485](#), [1486](#), [1497](#)
 - Administrative Install panel [1465](#), [1477](#)
 - Advanced Marimba Options dialog box [1471](#)
 - Altiris [1437](#)
 - Altiris Integration Panel [1465](#)
 - Altiris XML Template [1466](#)
 - creating administrative installations using [1436](#)
 - distributing packages to FTP servers [1437](#)
 - distributing packages to network locations using [1443](#)
 - distributing packages using Configuration Manager Web Console [1497](#)
- Distribution Output panel [1486](#)
- Distribution Summary panel [1486](#)
- Distribution Type panel [1464](#)
- FTP Location panel [1467](#)
- Installation Package panel [1485](#)
- LANDesk Integration Panel [1467](#)
- ManageSoft Package and Environmental Settings Panel [1439](#), [1468](#)
- Marimba Integration panel [1469](#), [1470](#)
- Marimba Patch Information Panel [1469](#)
- Network Location panel [1471](#)
- preparing for LANDesk distribution [1439](#)
- preparing for ManageSoft distribution [1439](#), [1468](#)
- preparing packages for distribution using [1435](#)
- preparing packages for Microsoft Configuration Manager [1497](#)
- preparing packages for SMS [1444](#)
- preparing packages for Tivoli [1446](#)
- preparing packages for ZENworks desktop application distribution [1452](#)
- preparing packages for ZENworks server distribution [1456](#)
- script-based setup.exe [1461](#)
- SMS Distribution Panel [1472](#)

- SPD Parameters panel [1473](#), [1474](#)
- Tivoli Integration panel [1472](#)
- Tivoli Settings panel [1472](#)
- using Marimba Channel Publishing [1441](#)
- Welcome panel [1463](#)
- ZENworks Configuration Management [1447](#), [1486](#)
- ZENworks Desktop Application Panel [1480](#)
- ZENworks Server Distribution - Distributor Panel [1483](#)
- ZENworks Server Distribution - Object Panel [1482](#)
- Distribution Wizard for ZENworks Configuration Management
 - ZENworks Configuration Management [1447](#), [1486](#)
- Division [1370](#)
- DLL Hell [1353](#)
 - avoiding [1353](#)
- DLL Information tab [1231](#)
- DLL-Based ACEs panel [1258](#)
 - Rules Wizard [1258](#)
- dual-mode patches [1335](#)
- duplicate package names [275](#)
 - during auto import [275](#)
- Duplicate Package options [1239](#)
 - ConflictSolver [1239](#)
- Duplicate Package tab [1239](#)
 - in ConflictSolver Options dialog [1239](#)
- DuplicateFileData ACE [1171](#)
- Dynamic Dependency Scanner [551](#)
- dynamic suite composition [810](#)
- Dynamic Suite Composition page [832](#)

E

- Edit Registry Key dialog [628](#)
- Edit Registry Key Dialog Box [628](#)
- Edit Windows Installer Package option [668](#)
- Edit Windows Installer Package panel [705](#)
 - Windows Installer Converter [705](#)
- editing packages [1056](#)
 - Direct Editor [1056](#)
- EnablePathVariables [628](#)
- engines [443](#)
 - including InstallScript engines with installation [443](#)
- enhancement
 - Customer Experience Improvement Program (CEIP) [148](#)
- enhancing packages [1001](#)
- ensuring package quality [1373](#)
- Enterprise Server Web tools
 - evaluating [73](#)
- error C1083 [1189](#)
- error control level [1042](#)
 - setting [1042](#)
- Evaluate package(s) for virtualization suitability [650](#), [700](#)
- evaluating
 - AdminStudio Enterprise Edition client tools [74](#)
 - AdminStudio Enterprise Server Web tools [73](#)
 - AdminStudio's Microsoft App-V support [75](#)
 - batch convert option [75](#)
- evaluation files [1016](#), [1020](#)
- evaluation license file
 - obtaining [1643](#)
- Exclude Element [681](#)
- Excluded Processes dialog box [514](#)
- excluding files [558](#)
- exclusion list [635](#)
 - specifying directory [114](#)
- exclusion lists [1390](#)
- Exclusion view [339](#)
- exclusions [558](#), [559](#), [560](#), [561](#), [562](#), [563](#), [564](#), [566](#), [567](#), [568](#), [569](#), [570](#), [571](#), [572](#), [622](#), [623](#), [624](#), [625](#)
 - all files in a directory [559](#)
 - all shortcuts in a directory [561](#)
 - anti-virus software [572](#)
 - configuring in Repackager [558](#)
 - directories and subdirectories [559](#)
 - directory [567](#)
 - editing existing file [567](#)
 - editing existing INI file [570](#)
 - editing existing registry [571](#)
 - external configuration file in Repackager [562](#)
 - file [566](#)
 - files [623](#)
 - INI file [568](#)
 - INI file sections [560](#), [569](#)
 - INI files [560](#), [624](#)
 - modifying Repackager external configuration file [563](#)
 - OS snapshot [563](#)
 - OS Snapshot Wizard [564](#)
 - OS Snapshot Wizard global [564](#)
 - project [563](#)
 - registry [625](#)
 - registry data [571](#)
 - registry keys [559](#)
 - registry values [560](#)
 - removing existing file [568](#)
 - removing existing INI file [570](#)
 - removing registry [572](#)
 - Repackager [564](#)
 - Repackager file [558](#)
 - Repackager global [564](#)
 - repackaging [563](#)
 - resetting to default values [622](#)

- shortcuts [561](#)
- shortcuts from subdirectories [561](#)
- specific file extensions [566](#)
- with specific file extensions [566](#)
- Exclusions Editor [563](#), [564](#), [566](#), [567](#), [568](#), [569](#), [570](#), [571](#), [572](#), [622](#), [623](#), [624](#), [625](#), [626](#), [627](#), [628](#)
 - About dialog box [628](#)
 - Choose Registry Key dialog box [627](#)
 - Directories and Files Excluded During Analysis dialog box [626](#)
 - directory exclusions [567](#)
 - Edit Registry Key dialog box [628](#)
 - editing existing file exclusions [567](#)
 - editing existing INI file exclusions [570](#)
 - editing existing registry exclusions [571](#)
 - excluding INI file sections [569](#)
 - File Exclusion Information dialog box [626](#)
 - files [566](#)
 - Files tab [623](#)
 - INI File Exclusion Information dialog [627](#)
 - INI files [568](#)
 - INI Files tab [624](#)
 - Keys Excluded During Registry Analysis dialog box [627](#)
 - menus [622](#)
 - OS Snapshot Wizard exclusions [564](#)
 - reference [622](#)
 - registry data [571](#)
 - Registry tab [625](#)
 - removing existing file exclusions [568](#)
 - removing existing INI file exclusions [570](#)
 - removing registry exclusions [572](#)
 - Repackager exclusions [564](#)
 - resetting exclusions to default values [622](#)
 - specific file extensions [566](#)
- Exclusions tab [594](#)
- experience
 - Customer Experience Improvement Program (CEIP) [148](#)
- expiration [889](#), [921](#)
- Expiration Message [924](#)
- Expire Period [923](#)
- Expression Builder dialog [1236](#)
- extended attribute options [1241](#)
 - ConflictSolver [1241](#)
- Extended Attribute Property dialog [350](#)
- extended attributes [224](#), [225](#), [227](#)
 - and Workflow Manager [227](#)
 - description file in Application Manager [225](#)
 - in Application Manager [224](#)
 - using in Application Manager [225](#)
- Extended Attributes tab [1241](#)
 - in ConflictSolver Options dialog [1241](#)

- extended attributes tables [408](#)
- Extended Attributes view [321](#), [326](#), [329](#), [330](#)
 - Application Manager [321](#), [326](#), [329](#), [330](#)
- external configuration file [562](#), [563](#)
 - modifying Repackager [563](#)
 - specifying in Repackager [562](#)
- ExtraHKCRPermanent [628](#)

F

- F [426](#)
- failsafe modes
 - setting up [1663](#), [1664](#)
- FAQ [132](#)
- fatal error C1083 [1189](#)
- feature [1021](#)
 - changing visibility [1021](#)
 - setting initial state of [1021](#)
- feature advertisement [1024](#)
- feature block 1 [846](#)
- feature block 2 [846](#)
- feature blocks [814](#)
- FEATURE lines
 - overview of [1673](#)
- feature properties [1023](#)
- features [1022](#)
 - editing description for [1022](#)
- Features view [1022](#), [1073](#)
 - Description property [1022](#)
- FILE [789](#), [824](#)
- file associations [1385](#)
 - checking [1385](#)
- File Associations view [1422](#)
- file conflicts [1118](#)
- File Element [681](#)
- File Exclusion Information dialog box [626](#)
- file exclusions [558](#), [567](#), [568](#)
 - editing existing [567](#)
 - removing existing [568](#)
 - Repackager [558](#)
- file extension conflicts [1118](#)
- File Isolation Options dialog box [993](#)
- File Locations tab [1064](#)
- files [1025](#), [1026](#), [1027](#)
 - adding [1025](#)
 - displaying from base Windows Installer package [1026](#)
 - excluding [558](#)
 - exclusion of specific extensions [566](#)
 - exclusions [566](#), [623](#)
 - preventing installation of from the Windows Installer package [1026](#)

- removing added [1027](#)
- storing in CAB in Tuner [1027](#)
- Files & Folders page [898](#)
 - creating a new folder [795](#), [866](#)
 - deleting files and folders from a ThinApp application [867](#)
 - deleting files and folders from an App-V application [796](#)
- files and folders [1025](#)
- Files and Folders view [599](#), [1025](#), [1026](#), [1027](#), [1075](#)
 - adding files in [1025](#)
 - displaying files from the base Windows Installer package [1026](#)
 - preventing installation of files from the Windows Installer package [1026](#)
 - removing added files from [1027](#)
- Files page [826](#)
- Files Report [1132](#), [1134](#)
- Files tab [623](#)
 - Exclusions Editor [623](#)
- Files view [340](#)
- Files/Components view [322](#)
- Filter view [1313](#)
- filters
 - deleting in PackageExpert [1286](#), [1290](#), [1313](#)
 - editing in PackageExpert [1285](#), [1289](#), [1313](#)
- Find dialog [348](#)
- finding and replacing [1057](#)
 - using Direct Editor [1057](#)
- Fingerprint Element [681](#)
- Finishing INI File Import panel [1093](#)
 - Import REG File Wizard [1093](#)
- Finishing Registry Import panel [1094](#)
 - Import REG File Wizard [1094](#)
- FlexWrap
 - advanced license settings [1656](#)
 - applications to which it can be applied [1610](#)
 - basic license settings [1655](#)
 - benefits [1610](#)
 - checking the status of the license server [1649](#)
 - communication issues [1640](#)
 - components of [1666](#)
 - creating a new license [1651](#)
 - creating an options file [1660](#)
 - installation notes [1640](#)
 - installing FlexWrap license file [1642](#)
 - license parameters [1668](#)
 - license server overview [1638](#)
 - overview [1610](#)
 - overviewX [1629](#)
 - platform issues [1639](#)
 - policies and options [1654](#)
 - purposes of [1610](#)

- saving a license file [1661](#)
- setting up failsafe modes [1663](#)
- setting up license server failsafe modes [1664](#)
- setting up regular (license) failsafe modes [1663](#)
- specifying license parameters [1668](#)
- starting the license server [1645](#)
- stopping the license server [1649](#)
- terminology [1666](#)
- testing a wrapped application [1662](#)
- tracking projects with [1651](#)
- understanding license file FEATURE lines [1673](#)
- using a wrapped application [1616](#)
- wrapping an executable [1651](#)
- FlexWrap license server
 - checking the status [1649](#)
 - starting [1645](#)
 - stopping [1649](#)
- FlexWrap License Server Component [1637](#)
- Flicense file
 - installing [1642](#)
- Folder Isolation Options dialog box [844](#), [912](#), [994](#)
- foreign key [388](#)
- frequency [145](#)
- frequently asked questions [132](#)
- FTP Location panel [1467](#)
 - Distribution Wizard [1467](#)
- FTP server [1053](#)
 - copying installation to [1053](#)
- FTP servers [1437](#)
 - distributing packages to using Distribution Wizard [1437](#)
- Full isolation option [871](#)
- FullyIsolate Element [681](#)

G

- General Information panel [1254](#)
 - Rules Wizard [1254](#)
- General Information tab [1228](#)
- general options [1241](#)
 - ConflictSolver [1241](#)
- General tab [593](#), [1241](#)
 - in ConflictSolver Options dialog [1241](#)
- generic transforms [1015](#), [1064](#)
 - creating [1015](#), [1064](#)
- Get Latest Version [258](#)
- Global Exclusions Editor
 - exclusion list [114](#)
- Global Exclusions list [114](#)
- Group Properties dialog [350](#)
- Group view [319](#)
- groups [222](#), [223](#), [224](#)

Index

- adding in Application Manager [222](#)
- copying packages to multiple Application Manager [223](#)
- deleting [162](#)
- deleting from Application Manager [223](#)
- editing properties in Application Manager [223](#)
- moving in Application Manager [224](#)
- organizing products into in Application Manager [222](#)
- removing from Application Manager [223](#)
- Groups panel [375](#), [377](#)
 - Merge Wizard [377](#)
 - Package Auto Import Wizard [375](#)
- groups table [408](#)
- guest account [173](#), [174](#)
 - setting up [173](#)

H

- hashes file [665](#)
- hashes.txt [665](#)
- help
 - Help Library conventions [77](#)
- help files [126](#), [1385](#)
 - associating with tasks [126](#)
 - checking [1385](#)
- Help Files view [1422](#)
- Help Telephone [1086](#)
- Help URL [1086](#)
- Help view [1071](#)
- HTTP [789](#), [824](#)
- HTTPS [789](#), [824](#)

I

- I [426](#)
- ICE errors [1113](#)
 - deleting [1115](#)
 - suppressing [1114](#)
 - viewing after validation [1113](#)
- ICEs [1020](#)
- ICO file [657](#)
- icons [308](#), [311](#), [1203](#)
 - Merge Modules View [311](#)
 - Product View [308](#), [1203](#)
- IF [426](#), [435](#)
- IMM [426](#)
- Impacts View on Patches Tab [1226](#)
 - ConflictSolver [1226](#)
- import [1124](#)
 - checking for conflicts during [1124](#)
- Import Conflict Options panel [1092](#), [1093](#)
 - Import INI File Wizard [1092](#)
- Import REG File Wizard [1093](#)
- Import INI File panel [1092](#)
 - Import INI File Wizard [1092](#)
 - Import Conflict Options panel [1092](#)
 - Import INI File panel [1092](#)
 - Welcome panel [1092](#)
- import options [1242](#)
 - ConflictSolver [1242](#)
- Import REG File Wizard [1093](#), [1094](#)
 - Finishing INI File Import panel [1093](#)
 - Finishing Registry Import panel [1094](#)
 - Import Conflict Options panel [1093](#)
 - Import Registry File panel [1093](#)
 - Welcome panel [1093](#)
- Import Registry File panel [1093](#)
 - Import REG File Wizard [1093](#)
- Import Tab [316](#), [1208](#)
- Import tab [1242](#)
 - in ConflictSolver Options dialog [1242](#)
- Import Wizard [364](#), [365](#), [373](#)
 - Destination Group panel [373](#)
 - Summary panel [373](#)
 - Welcome panel [365](#)
- importing [230](#), [232](#), [234](#), [238](#), [244](#), [245](#), [246](#), [247](#), [248](#), [249](#), [436](#)
 - ad-hoc data [230](#)
 - Marimba NCP files [230](#), [247](#), [248](#)
 - merge modules [230](#), [244](#), [245](#)
 - merge modules and Windows Installer packages
 - simultaneously into Application Manager [436](#)
 - MSI packages [230](#), [232](#), [238](#)
 - OS patches using OS Security Patch Wizard [1328](#)
 - OS Snapshots [230](#), [246](#)
 - OS snapshots [245](#)
 - other setup types [249](#), [250](#)
 - specifying duplicate package identifiers in Application Manager [234](#)
 - transforms [238](#)
 - virtual packages into the Application Catalog [238](#)
 - Windows Installer packages [230](#), [232](#), [234](#)
- importing data
 - importing into the Software Repository [253](#)
 - performing ad-hoc import [236](#)
- including additional MSIs in virtual package [836](#), [882](#), [908](#)
- INF file [1335](#)
- INI file [1092](#)
 - Import INI File Wizard [1092](#)
- INI file actions [1035](#)
 - modifying [1035](#)
- INI File Changes view [323](#), [327](#), [330](#)

- INI file conflicts [1118](#)
- INI File Exclusion Information dialog box [627](#)
- INI file exclusions [570](#)
 - editing existing [570](#)
 - removing existing [570](#)
- INI file keys [1035](#), [1037](#)
 - modifying [1035](#)
 - removing [1037](#)
- INI file sections [560](#), [569](#)
 - excluding [560](#), [569](#)
- INI file values [1035](#)
 - modifying [1035](#)
- INI files [560](#), [568](#), [624](#), [1033](#), [1034](#), [1035](#), [1036](#), [1037](#)
 - adding [1034](#)
 - adding new keys [1035](#)
 - adding sections to [1035](#)
 - exclusions [560](#), [568](#), [624](#)
 - importing existing [1034](#)
 - removing [1036](#)
 - removing sections from [1037](#)
- INI Files tab [624](#)
 - Exclusions Editor [624](#)
- INI Files view [603](#), [1034](#), [1035](#), [1036](#), [1037](#), [1081](#)
 - adding INI files in [1034](#)
 - adding new INI file keys [1035](#)
 - adding sections to INI files in [1035](#)
 - importing existing INI files into [1034](#)
 - modifying INI file keys [1035](#)
 - removing INI file keys from [1037](#)
 - removing INI files from [1036](#)
 - removing sections from INI files from [1037](#)
- initial state [1021](#)
 - setting for features [1021](#)
- initial state property [1023](#)
- install monitoring [450](#), [452](#)
- installation [1052](#), [1053](#)
 - copying to FTP server [1053](#)
 - copying to network location [1052](#)
- Installation Designer
 - opening [770](#)
- installation monitoring [450](#), [452](#)
 - excluding processes from [514](#)
- Installation Package panel [1485](#)
 - Distribution Wizard [1485](#)
- installDir [516](#)
- installing
 - Novell ZENworks Desktop Management Agent [528](#)
- InstallLocation [145](#)
- InstallScript engines [443](#)
- InstallScript MSI [492](#)
 - converting [506](#)
 - converting to Basic MSI with InstallScript support [492](#)
 - converting to Repackager project [492](#)
- InstallScript MSI Conversion Output panel [509](#)
 - Repackaging Wizard [509](#)
- InstallScript MSI Identified panel [506](#)
 - Repackaging Wizard [506](#)
- InstallScript Scan [443](#), [492](#)
- InstallShield
 - about the virtualization Assistants [768](#)
 - App-V Assistant [771](#)
 - Citrix Assistant [935](#)
 - integration of App-V Assistant [768](#)
 - integration of virtualization Assistants [768](#)
 - ThinApp Assistant [850](#), [851](#)
- InstallShield 2008 Professional
 - compared to InstallShield Editor [1010](#)
- InstallShield Editor [550](#), [1001](#)
 - ConflictSolver tab on InstallShield Editor Options dialog [1009](#)
 - customizing and authoring packages [1001](#)
 - Dynamic Dependency Scanner [551](#)
 - editing generated Repackager projects [550](#)
 - features [1010](#)
 - integration with Application Catalogs [1004](#)
 - Options dialog [1009](#)
 - vs. InstallShield 2008 Professional [1010](#)
- InstallShield Editor project [532](#)
 - building in Repackager [532](#)
- InstallShield log files [531](#)
 - converting to Repackager project [531](#)
- InstallShield Professional [508](#), [616](#)
 - SmartScan Wizard [617](#)
- InstallShield Professional Setup panel [508](#)
 - Repackaging Wizard [508](#)
- InstallShield Professional setups [573](#)
 - scanning in Repackager [573](#)
- InstallShield SmartScan options [542](#), [614](#)
- Interface [111](#)
 - using AdminStudio [111](#)
- Interm directory [661](#), [857](#)
- internal consistency evaluators [1020](#)
- invalid Windows Installer package [1018](#)
 - handling [1018](#)
- is [516](#)
- ISCSAboutNew [1100](#)
- isolated components [550](#)
- Isolated Components Design dialog [1368](#)
- IsolatedComponents [1370](#)
- isolating applications [1353](#)
- Isolating Windows Installer Packages Using Application Isolation Wizard [551](#)

Index

isolation [550](#), [1353](#), [1356](#), [1359](#), [1361](#), [1362](#)
 applications [1359](#)
 filtering file listings when manually configuring [1362](#)
 methods [1356](#)
 modifying default recommendations [1361](#)
 reasons not to do [1353](#)
 reasons to do [1353](#)
Isolation Method panel [1364](#)
 Application Isolation Wizard [1364](#)
isolation methods [1356](#)
 assemblies and manifests [1356](#)
 Windows Installer isolated components [1356](#)
isolation options
 in ThinApp Assistant [870](#)
 inheritance from folders to files in a Citrix profile [960](#)
 inheritance from folders to files in a ThinApp application [872](#)
 inheritance from folders to files in an App-V application [802](#)
 inheritance in the registry in a Citrix profile [969](#)
 inheritance in the registry in a ThinApp application [880](#)
 inheritance in the registry in an App-V application [809](#)
 overview of [800](#), [869](#), [958](#)
 setting for folders and files in App-V Assistant [801](#)
 setting for folders and files in Citrix Assistant [960](#)
 setting for folders and files in ThinApp Assistant [871](#)
 setting in App-V Assistant [800](#)
 setting in Citrix Assistant [957](#)
 setting in registry for Citrix profile [968](#)
 setting in registry for ThinApp application [879](#)
 setting in ThinApp Assistant [869](#)
Isolation Options dialog box [588](#)
isolation tests [1394](#), [1395](#)
 performing [1394](#), [1395](#)
Isolation Tests view [1430](#)
isrepackager.context.ini [695](#)
ISRepackager.ini [635](#), [638](#)
isrepackager.ini [114](#)
ISRIsolation [555](#)
issnapshot.ini [379](#)

J

Job Details Report (Predeployment Test) [1588](#)
Job Manager [1753](#)
 about Templates [1771](#)
 available Job Steps [1755](#)
 configuration options [1775](#)
 conflict resolution and the Software Repository [1758](#)
 contextual packages [1757](#)
 creating a new Custom Job [1761](#)

 creating a new Job [1760](#)
 creating a new Template [1772](#)
 Custom Jobs [1757](#)
 deleting a Job [1770](#)
 deleting a Template [1775](#)
 dynamic group content identification [1757](#)
 editing a Job [1766](#)
 editing a Template [1774](#)
 email alerts [1768](#)
 evaluating [73](#)
 executing jobs [1779](#)
 Job Manager Engine [1777](#)
 rescheduling a Job [1767](#)
 selecting packages imported in previous Job Steps [1757](#)
 setting detection options [1775](#)
 tables [420](#)
 Template Job vs. Custom Job [1756](#)
 user permissions [1759](#)
 viewing Job Manager Engine status [1780](#)
 viewing Job status [1769](#)
Job Manager Engine [1777](#)
 adding an Application Catalog to [1778](#)
 deleting a connection to [1778](#)
 executing jobs [1779](#)
 opening [1778](#)
 setting conflict detection options [1781](#)
 setting validation options [1782](#)
 shutting down and restarting Application Catalog [1779](#)
 viewing status of in Job Manager [1780](#)
Job Summary Report (Predeployment Test) [1585](#)

K

KeyPath ACE [1173](#)
Keys Excluded During Registry Analysis dialog [627](#)

L

-L [426](#)
LANDesk distribution [1439](#)
 creating with Distribution Wizard [1439](#)
LANDesk Integration Panel [1467](#)
 Distribution Wizard [1467](#)
language requirements
 setting in Citrix profile [948](#)
Launch App-V Application dialog box [845](#)
LDAP attributes [183](#)
 Active Directory [183](#)
Legacy Client [1499](#)
legacy setups [443](#), [458](#), [526](#)
 converting [526](#)

- repackaging [443](#), [458](#)
- Legacy Upgrade Wizard
 - upgrading legacy Application Catalogs [221](#), [386](#)
- license
 - advanced settings [1656](#)
 - basic settings [1655](#)
 - saving a wrapped application [1661](#)
- License Agreement dialog [1045](#)
 - suppressing [1045](#)
- license file
 - creating in FlexWrap [1651](#)
 - obtaining an evaluation license [1643](#)
 - saving [1661](#)
- license parameters [1668](#)
 - specifying [1668](#)
- license server
 - checking the status of [1649](#)
 - overview [1638](#)
 - starting [1645](#)
 - stopping [1649](#)
- LimitedUI [628](#)
- limiting tool accessibility [122](#)
- load order group [1042](#)
 - setting [1042](#)
- Location panel [1094](#)
 - Packaging Wizard [1094](#)
- lockdown and runtime tests [1393](#), [1395](#)
 - performing [1393](#)
 - running in restricted environments [1395](#)
- Lockdown and Runtime Tests view [1425](#)
- lockdown testing [1392](#)
- log file [437](#)
 - creating during command-line import [437](#)
- Log Monitor [883](#)
- Log Monitor tracing options [883](#)
- logging in to AdminStudio Enterprise Server [172](#), [173](#)
 - anonymous [173](#)
 - anonymous account [174](#)
 - automatically [172](#)
 - forgetting your password [172](#)
- logging in to Workflow Manager
 - guest account [173](#), [174](#)

M

- Machine Result Summary Report (Predeployment Test) [1593](#)
- Management Points [1499](#)
- ManageSoft distribution [1439](#), [1468](#)
 - creating with Distribution Wizard [1439](#), [1468](#)
- ManageSoft Package and Environmental Settings Panel
 - [1439](#), [1468](#)

- manifest
 - in Citrix profile [665](#)
- Manifest and Assembly Design dialog [1368](#)
- Manifest file [657](#)
- manifest files [550](#)
- manifest options [1366](#)
 - Application Isolation Wizard [1366](#)
- Manifest Options tab [1366](#)
 - Advanced Options dialog [1366](#)
- Manifests [1370](#)
- manifests [552](#), [1357](#), [1372](#)
 - application [552](#), [1357](#)
 - assembly [552](#), [1357](#)
 - checking in QualityMonitor [1388](#)
 - checking with QualityMonitor
 - QualityMonitor
 - checking manifests [1388](#)
 - examples of [1372](#)
 - testing [1388](#)
- Manifests view [1423](#)
- Manual Test Perform dialog box [1319](#)
- Manual Test Resolve dialog box [1320](#)
- manual tests
 - about [1294](#)
 - adding [1294](#)
 - creating in PackageExpert [1296](#)
 - Manual Test Perform dialog box [1319](#)
 - Manual Test Resolve dialog box [1320](#)
 - resolving [1297](#), [1298](#)
 - running [1297](#)
 - sample [1299](#)
- Mapped Network Drive Changes go to Sandbox [897](#)
- Marimba [1441](#)
 - patch information [1469](#)
 - using channel publishing in Distribution Wizard [1441](#)
- Marimba advanced options [1471](#)
- Marimba Advanced Options dialog box [1471](#)
- Marimba Integration panel [1469](#), [1470](#)
 - Distribution Wizard [1469](#), [1470](#)
- Marimba NCP File View [329](#), [1221](#)
- Marimba NCP files [230](#), [247](#)
 - importing [230](#), [247](#), [248](#)
 - manifest.ncp [248](#)
- Marimba Patch Information Panel [1469](#)
 - Distribution Wizard [1469](#)
- menus [140](#), [305](#), [584](#), [622](#), [1059](#), [1200](#), [1406](#)
 - AdminStudio [140](#)
 - Application Manager [305](#)
 - Application Manager context [312](#)
 - ConflictSolver [1200](#)
 - ConflictSolver context [1206](#)

Index

- Exclusions Editor [622](#)
- QualityMonitor [1406](#)
- Repackager [584](#)
- Merge Module view [326](#), [338](#), [1219](#)
- merge modules [244](#), [436](#)
 - importing [230](#), [244](#), [245](#)
 - importing multiple using a configuration file [436](#)
 - importing simultaneously with Windows Installer packages [436](#)
- Merge Modules View [311](#)
 - Application Manager [311](#)
 - icons [311](#)
- Merge Modules view [324](#)
- merge replication
 - using SQL Server [296](#)
- Merge Wizard [376](#), [377](#)
 - Groups panel [377](#)
 - Progress panel [377](#)
 - Source Application Catalog panel [377](#)
 - Welcome panel [377](#)
- Merged isolation option [870](#)
- merging Application Catalog information [264](#)
- Message Element [681](#)
- Method Selection panel [501](#)
 - advanced settings [514](#)
 - Repackaging Wizard [501](#)
- metrics [1192](#)
 - viewing for ACEs [1192](#)
- Microsoft Application Virtualization (App-V). See App-V applications.
- Microsoft Application Virtualization. See App-V applications.
- Microsoft App-V applications. See App-V applications.
- Microsoft App-V Assistant. See App-V Assistant.
- Microsoft App-V Sequencer [775](#), [783](#)
 - compared with App-V Assistant [775](#)
 - interface comparison to App-V Assistant [783](#)
 - quick start guide [783](#)
- Microsoft Configuration Manager [1498](#)
- Microsoft Configuration Manager Server [1499](#)
- Microsoft patches
 - Consolidated Patch Report [1343](#)
 - performing patch impact analysis [1336](#)
 - Properties dialog box [1350](#), [1351](#), [1352](#)
 - viewing information in ConflictSolver [1339](#)
- Microsoft System Center Configuration Manager [1498](#)
 - integration with Configuration Manager Web Console [1498](#)
- MIF [1055](#)
 - instructing SMS to create file [1055](#)
- mm [516](#)
- mode [516](#)

- Modify button [1050](#)
 - disabling in Add/Remove Programs [1050](#)
- modifying INI file actions [1035](#)
- modifying INI file keys [1035](#)
- modifying INI file values [1035](#)
- MoveFileData ACE [1174](#)
- mp [516](#)
- ms [516](#)
- MSI command-line options [1047](#)
 - when to use Dialogs view instead [1047](#)
- MSI Doctor [1397](#)
 - install or configure products or features [1400](#)
 - reinstall components [1402](#)
 - reinstall features [1401](#)
 - verify files [1399](#)
 - view properties [1399](#)
- MSI packages [232](#), [234](#), [1117](#), [1131](#)
 - building in Repackager [532](#)
 - handling invalid [1018](#)
 - handling invalid in ConflictSolver [1117](#)
 - importing [230](#), [232](#)
 - prevalidating [1017](#)
 - resolving conflicts directly within [1131](#)
- MSI Source Information panel [1250](#), [1259](#)
 - Conflict Wizard [1250](#)
 - Validation Wizard [1259](#)
- msi2vp.exe [693](#)
- MultiUserShortcuts [628](#)

N

- NCP files [247](#), [248](#)
 - importing [247](#), [248](#)
 - Marimba NCP File View [329](#), [1221](#)
 - NCP View [329](#), [1221](#)
- nested .msi custom action [639](#)
- network location [1052](#)
 - copying installation to [1052](#)
- Network Location panel [1471](#)
 - Distribution Wizard [1471](#)
- network locations [1443](#)
 - distributing packages to using Distribution Wizard [1443](#)
- New Workflow Project Wizard [127](#), [153](#), [154](#)
 - creating workflows with [127](#)
 - reference [153](#)
 - Source Package panel [154](#)
 - Target Directory and File Name panel [154](#)
 - Welcome panel [153](#)
 - Workflow Selection panel [154](#)
- New Workflow Project wizard [129](#)
 - example using [129](#)

- new workflows [127](#)
 - creating [127](#)
- NewComponents [1370](#)
- notes [125](#)
 - creating for tasks [125](#)
- Novell ZENworks Desktop Management Agent
 - installing [528](#)
- Novell ZENworks projects [527](#)
 - .aot and .axt files [527](#)
 - application object template files [528](#)
 - converting to Repackager project [527](#)
- NT service arguments [1040](#)
- NT service dependencies [1041](#)
 - setting [1041](#)
- NT service description [1041](#)
 - setting [1041](#)
- NT service display name [1041](#)
 - setting [1041](#)
- NT service type arguments [1040](#)
- NT services [1040](#)
- NT services conflicts [1118](#)
- NT Services view [1040](#), [1041](#), [1042](#), [1043](#), [1083](#)
 - setting the NT service arguments in [1040](#)
 - setting the NT service dependencies in [1041](#)
 - setting the NT service description in [1041](#)
 - setting the NT service display name in [1041](#)
 - setting the NT service error control level in [1042](#)
 - setting the NT service load order group in [1042](#)
 - setting the NT service overall install result in [1042](#)
 - setting the NT service start name and password in [1043](#)
 - setting the NT service start type in [1043](#)
 - setting the NT service type in [1043](#)

O

- o [516](#)
- ODBC data source attributes [1038](#), [1039](#), [1040](#)
 - adding new [1038](#)
 - editing [1039](#)
 - removing [1040](#)
- ODBC data sources [1039](#), [1389](#)
 - checking [1389](#)
 - removing existing [1039](#)
- ODBC Data Sources view [1424](#)
- ODBC driver attributes [1038](#), [1039](#)
 - adding new [1038](#)
 - editing [1039](#)
 - removing [1039](#)
- ODBC drivers [1390](#)
 - checking [1390](#)
- ODBC Drivers view [1424](#)
- ODBC resource conflicts [1118](#)
- ODBC resources [1037](#)
- ODBC Resources view [1038](#), [1039](#), [1040](#), [1082](#)
 - adding new data sources [1038](#)
 - adding new ODBC data source attributes [1038](#)
 - adding new ODBC driver attributes [1038](#)
 - editing ODBC data source attributes [1039](#)
 - editing ODBC driver attributes [1039](#)
 - removing existing ODBC data sources from [1039](#)
 - removing ODBC data source attributes from [1040](#)
- of [516](#)
- Open a recent transform file option [1070](#)
- Open an existing transform file option [1071](#)
- Open QualityMonitor Project dialog [1413](#)
- Optimize for Offline Use [815](#)
- Optimize for Streaming [815](#)
- opt-in or opt-out of Customer Experience Improvement Program (CEIP) [148](#)
- options
 - FlexWrap [1654](#)
- Options dialog [145](#), [591](#), [1237](#), [1239](#), [1241](#), [1242](#), [1243](#), [1244](#), [1414](#)
 - Conflicts tab in ConflictSolver [1237](#)
 - Duplicate Package tab in ConflictSolver [1239](#)
 - Extended Attributes tab in ConflictSolver [1241](#)
 - General tab in ConflictSolver [1241](#)
 - Import tab in ConflictSolver [1242](#)
 - in AdminStudio [145](#)
 - Resolution tab in ConflictSolver [1243](#)
 - Validate tab in ConflictSolver [1244](#)
- Options dialog box [1064](#)
- options file
 - creating in FlexWrap [1660](#)
- Options.ini file [628](#), [635](#)
- options.ini file
 - OtherComponentFileExtensions [628](#)
 - support for user-defined extensions [628](#)
- Original InstallShield Professional Setup panel [616](#)
 - Repackaging Wizard [616](#)
- OS requirements [612](#)
 - option to match legacy system's OS requirements for SmartScan [612](#)
- OS Security Patch Wizard [1325](#), [1328](#)
 - about missing patch files [1335](#)
 - dual-mode patches [1335](#)
- OS Security Patch Wizard Tables [425](#)
- OS Snapshot
 - display limits [1241](#)
 - Shortcuts view [328](#)
 - tables [401](#)
- OS snapshot [563](#)

- exclusions [563](#)
- OS Snapshot Summary panel [379](#)
 - OS Snapshot Wizard [379](#)
- OS Snapshot view [326](#), [1219](#)
- OS Snapshot Wizard [302](#), [303](#), [304](#), [378](#), [379](#), [564](#)
 - Analysis Options dialog [379](#)
 - best practices [302](#)
 - concept [302](#)
 - configuring analysis options [303](#)
 - exclusions [564](#)
 - OS Snapshot Summary panel [379](#)
 - performing OS Snapshots [304](#)
 - Performing Snapshot panel [378](#)
 - Project Information panel [378](#)
 - reference [378](#)
 - Welcome panel [378](#)
- OS Snapshots
 - importing [230](#), [246](#)
- OS snapshots [224](#), [245](#), [302](#)
 - capturing [302](#)
 - importing [245](#)
 - moving in Application Manager [224](#)
 - taking [302](#)
- OSD [329](#), [1221](#)
 - Marimba NCP File View [329](#), [1221](#)
 - NCP View [329](#), [1221](#)
- OSD file [657](#)
 - editing [658](#)
- other setup types [250](#)
 - importing [249](#), [250](#)
- Other Setup Types View [328](#), [1220](#)
- OtherComponentFileExtensions [628](#)
- OtherFilesNewComponents [628](#)
- Output Tab [316](#), [1208](#)
- Output Window [316](#), [1063](#), [1208](#)
 - Conflicts Tab [316](#), [1208](#)
 - Import Tab [316](#), [1208](#)
 - Output Tab [316](#), [1208](#)
 - Package Auto Import Tab [316](#), [1208](#)
 - Patch Impact Tab [316](#), [1208](#)
 - Search Results Tab [316](#), [1208](#)
 - Validate Tab [316](#), [1208](#)
- output window
 - in PackageExpert [1308](#)
- overall install result [1042](#)
 - setting [1042](#)

P

- P [426](#)
- package [1017](#), [1018](#), [1504](#)

- distributing to Configuration Manager Server using
 - Configuration Manager Web Console [1522](#)
- distributing using Configuration Manager Web Console [1504](#)
- handling invalid Windows Installer [1018](#)
- modifying distribution settings in Configuration Manager Web Console [1522](#)
- prevalidating MSI [1017](#)
- saving a package on Configuration Manager Server using Configuration Manager Web Console [1522](#)
- selecting for distribution using Configuration Manager Web Console [1504](#)
- Package Administration Page (Configuration Manager Web Console) [1545](#)
- package and transform file tables [408](#)
- Package Auto Import [265](#), [266](#), [269](#), [270](#), [271](#), [272](#)
 - Application Manager [265](#), [266](#), [272](#)
 - deleting a link to a network directory [275](#)
 - deleting remote Application Catalog link [271](#)
 - duplicate package names during import [275](#)
 - editing a link to a network directory [274](#)
 - editing in Application Manager [269](#)
 - forcing an import of packages in a network directory [275](#)
 - in Application Manager [269](#), [270](#), [271](#)
 - linking to a network directory [272](#)
 - refreshing linked packages in Application Manager [270](#)
- Package Auto Import Properties Dialog [352](#), [354](#)
- Package Auto Import Tab [316](#), [1208](#)
- Package Auto Import Wizard [271](#), [373](#), [374](#), [375](#), [376](#)
 - Folder Panel [375](#)
 - Groups panel [375](#), [376](#)
 - Monitoring Process Selection panel [374](#)
 - Remote Link Application Catalog panel [374](#)
 - Summary panel [376](#)
 - Welcome panel [374](#)
- package configuration options [1506](#)
 - access accounts [1510](#)
 - advertisements [1519](#)
 - distribution points [1511](#)
 - package settings [1506](#)
 - programs [1512](#)
 - setting in Configuration Manager Web Console [1505](#)
- package content [1025](#)
 - configuring [1025](#)
- package conversion options [541](#), [612](#)
- Package Conversion panel
 - Windows Installer Converter [706](#)
- package definition file [1053](#), [1527](#)
 - Configuration Manager Web Console [1527](#)
 - creating [1053](#)
- package definition file (PDF) [1527](#)

- Package Information page [821](#)
- Package Optimizations
 - Optimize for Offline Use [815](#)
 - Optimize for Streaming [815](#)
- package optimizations [814](#)
- Package Optimizations dialog box [845](#)
- package quality [1373](#)
- Package Report [1132](#), [1134](#)
- package settings [1506](#)
- Package Summary panel [1095](#)
 - Packaging Wizard [1095](#)
- Package Validation view [1017](#), [1071](#)
 - prevalidating a Windows Installer package [1017](#)
- Package view [1052](#), [1053](#), [1089](#)
 - deploying to FTP server [1053](#)
 - deploying to network location [1052](#)
- Package.DAT [660](#), [776](#), [855](#)
- package.ini [925](#)
- PackageExpert
 - about [1262](#)
 - ad-hoc test results [1290](#)
 - automatically resolving errors [1293](#)
 - automatically running PackageExpert tests during Repackager build process [538](#), [609](#)
 - changing the severity of a message [1283](#)
 - changing the severity of a specific error [1282](#)
 - choosing .mst or editing .msi to resolve errors [1265](#)
 - Configuration view [1312](#)
 - configuring [1264](#)
 - Connect Application Catalog dialog box [1314](#)
 - creating a manual test [1296](#)
 - Custom Test Results Editor dialog box [1317](#)
 - customizing test results [1282](#)
 - deleting a package's local test results [1281](#)
 - deleting a package's published test results [1282](#)
 - deleting a result filter [1286](#), [1290](#), [1313](#)
 - deleting a test result message [1293](#)
 - deleting test results [1281](#)
 - editing a filter [1285](#), [1289](#), [1313](#)
 - editing a test result message [1292](#)
 - excluding test results [1287](#), [1288](#)
 - Filter view [1313](#)
 - filtering [1287](#), [1288](#)
 - Home Page [1302](#)
 - icons [1305](#), [1306](#)
 - interface [1302](#)
 - managing test results [1271](#)
 - Manual Test Perform dialog box [1319](#)
 - Manual Test Resolve dialog box [1320](#)
 - message severity icons [1305](#)
 - Options view [1313](#)
 - output window [1308](#)
 - package tree [1303](#)
 - package tree icons [1306](#)
 - prompting to publish test results [1279](#)
 - publishing test results to the Application Catalog [1278](#)
 - reference [1301](#)
 - resolving a manual test [1297](#), [1298](#)
 - resolving errors [1293](#)
 - run status [1308](#)
 - running a manual test [1297](#)
 - sample manual test [1299](#)
 - Select a Package dialog box [1321](#)
 - setting default test configuration [1264](#)
 - setting default tests [1264](#)
 - setting defaults [1264](#)
 - setting resolution options [1264](#)
 - specifying the resolution method [1265](#)
 - status bar [1307](#)
 - suppressing the display of a message [1287](#), [1288](#)
 - Test & Resolve view [1309](#)
 - Test Result Filter Editor dialog box [1322](#)
 - Test Results view in ConflictSolver [1217](#), [1269](#)
 - testing external packages [1267](#)
 - testing packages [1266](#)
 - testing packages from a local file system [1267](#)
 - testing packages in an Application Catalog [1268](#)
 - testing packages in ConflictSolver [1269](#)
 - toggling between local and published test results [1279](#)
 - toolbar menus [1304](#)
 - Unpublished Packages dialog box [1323](#)
 - viewing Catalog or Local results [1279](#)
 - viewing test results [1271](#), [1272](#), [1273](#)
 - viewing test results in ConflictSolver [1280](#)
- packages [223](#), [230](#), [1011](#), [1052](#), [1056](#), [1132](#), [1435](#), [1502](#)
 - copying to multiple Application Manager groups [223](#)
 - customizing [1011](#)
 - deleting from Application Catalog [230](#)
 - distributing with Configuration Manager Web Console [1502](#)
 - editing with Direct Editor [1056](#)
 - preparing for distribution [1435](#)
 - preparing in Tuner for distribution [1052](#)
 - reimporting into ConflictSolver [1132](#)
 - testing prior to deployment [1558](#)
 - version management [256](#)
- packaging [1094](#)
 - Packaging Wizard [1094](#)
- Packaging Wizard [1052](#), [1053](#), [1094](#), [1095](#)
 - deploying to FTP server [1053](#)
 - deploying to network location [1052](#)
 - Location panel [1094](#)

Index

- Package Summary panel 1095
- Setup.exe panel 1095
- SMS panel 1095
- parameters 1056
 - Setup.ini 1056
- password 172, 1043
 - forgetting your Workflow Manager password 172
 - resetting your Workflow Manager password 172
 - setting NT service 1043
- Password Required dialog 592
- patch 435
 - applying to package during command-line import 435
- patch file
 - associating with package 250
 - performing an ad-hoc import 250
- patch files
 - performing ad-hoc import 236
- patch impact analysis 1336
- Patch Impact Analysis Wizard 1325, 1336, 1344
 - OS Snapshots panel 1346
 - Source Patches panel 1345
 - Summary Information panel 1346
 - Target Products panel 1346
 - Welcome panel 1344
- Patch Impact Tab 316, 1208
- Patch Impact view 1217
- Patch View 1225
- patches 238
 - dual-mode 1335
 - Patches Group View 334
 - Patches View 334, 336, 1222, 1225
 - performing patch impact analysis 1336
 - viewing information in ConflictSolver 1339
 - viewing patch impact analysis results 1343
- Patches Group View 334
- Patches Tab
 - ConflictSolver 1226
- Patches tab 334, 1222, 1339
 - Patches Group View 334
 - Patches View 336, 1225
- Patches View 336, 1225
- pc 516
- PDF (package definition file) 1527
 - using in Configuration Manager Web Console 1527
- pdt 388
 - Predeployment Test Tables 426
- People Administration Page 176
- performing OS Snapshots 304
- Performing Snapshot panel 378
 - OS Snapshot Wizard 378
- PermanentSystemFiles 628
- PermanentSystemFilesSubfolders 628
- permissions 122, 131, 194
 - assigning to a publication 290
 - copying roles 198
 - creating new roles 197
 - database server 389
 - deleting roles 198
 - Job Manager 1759
 - limiting tool accessibility 122
 - minimum for AdminStudio 389
- persisted conflicts 1126
 - deleting 1126
- persistence 1126
- person
 - disabling their user account 161
- Person Details View 178
- policies
 - FlexWrap 1654
- ports
 - port 443 (HTTPS) 73
 - port 80 (HTTP) 73
 - port 8443 73
 - used in activation 73
- PostReplication.sql 283
- postvalidation 1016, 1019
- Postvalidation view 1019, 1088
 - postvalidating transforms 1019
- pp 516
- Pre-Conversion Evaluation 673
 - rules 677
- predefined folders
 - controlling display of in App-V Assistant 796
 - controlling display of in Citrix Assistant 956
 - controlling display of in ThinApp Assistant 867
- predeployment analysis 1558
- Predeployment Test 1551, 1558, 1569, 1570, 1571, 1572, 1573, 1574, 1583, 1584, 1585, 1588, 1593, 1597
 - about 1552
 - assigning ASP.NET permissions 1597
 - assigning permissions to Application Catalogs 1597
 - assigning permissions to Microsoft Access databases 1597
 - changes made during test-ready package creation 1582
 - configuring Predeployment Test Reports Web site 1567, 1574
 - configuring Reports Web site 1558
 - ConflictSolver ACE reference 1601
 - Connection to New Catalog Page 1597
 - creating a test-ready package 1558
 - custom properties 1576
 - distributing test-ready package 1562

- Job Details Report [1588](#)
- Job Summary Report [1585](#)
- Machine Result Summary Report [1593](#)
- performing analysis [1562](#)
- Predeployment Test Preparation Wizard [1569](#)
- Predeployment Test Reports Web site [1567](#)
- Predeployment Test Results By Job Report [1584](#)
- Predeployment Test Results By Machine Report [1593](#)
- reference [1567](#)
- reports [1583](#)
- sample XML files [1577](#)
- setting command line parameters [1560](#)
- Start Page [1568](#)
- Test Details Report [1589](#)
- test identifiers [1576](#)
- Test Result Summary Report [1586](#)
- testing [1562](#)
- viewing reports [1564](#)
- wizard [1569](#)
- Predeployment test
 - wizard [1569](#)
- Predeployment Test Preparation Wizard [1558](#), [1569](#)
 - setting command line parameters [1560](#)
- Predeployment Test Reports Web site [1574](#)
 - configuring [1558](#)
- Predeployment Test Results By Job Report (Predeployment Test) [1584](#)
- Predeployment Test Results By Machine Report (Predeployment Test) [1593](#)
- PreReplication.sql [283](#)
- prevalidation [1016](#), [1017](#), [1018](#)
 - handling invalid packages [1018](#)
- Prevalidation view [1071](#)
- primary application directory [798](#)
 - explicitly set [798](#)
 - location of shortcut in ProgramFilesFolder [798](#)
 - location of shortcut not in ProgramFilesFolder [798](#)
 - ProgramFilesFolder [798](#)
 - value of INSTALLDIR variable [798](#)
- Primary Application Directory dialog box [847](#)
- primary key [388](#)
- privacy
 - Customer Experience Improvement Program (CEIP) [148](#)
- private key [1359](#)
- procedures and tasks [1021](#), [1025](#), [1026](#), [1027](#), [1028](#), [1029](#), [1030](#), [1031](#), [1032](#), [1033](#), [1034](#), [1035](#), [1036](#), [1037](#), [1038](#), [1039](#), [1040](#), [1041](#), [1042](#), [1043](#), [1044](#), [1045](#), [1046](#), [1048](#), [1049](#), [1050](#), [1051](#), [1068](#), [1070](#), [1071](#)
- Process Assistant [82](#), [84](#), [85](#), [86](#), [87](#), [88](#), [89](#), [90](#), [91](#), [92](#), [93](#), [94](#), [99](#), [100](#), [101](#), [102](#), [103](#), [104](#), [105](#), [106](#), [107](#), [108](#), [109](#), [110](#), [444](#)
- Assistant Tab [136](#)
- Best Practices Page [95](#)
- product code conflicts [1118](#)
- Product Info and Update URL [1086](#)
- Product Information view [1420](#)
- Product Name Change dialog [275](#)
- product properties [1022](#), [1023](#)
 - default destination [1022](#)
 - default organization [1022](#)
 - destination variable [1023](#)
- Product Properties view [1022](#), [1023](#), [1072](#)
 - changing destination variable [1023](#)
 - setting the default destination [1022](#)
 - setting the default organization [1022](#)
- Product View [271](#), [308](#), [1203](#)
 - Application Manager [308](#)
 - ConflictSolver [1203](#)
 - icons [308](#), [1203](#)
- Product view [319](#), [1211](#)
- products
 - moving in Application Manager [224](#)
 - organizing into groups in Application Manager [222](#)
- Products view [340](#)
- Profile Files page [981](#)
 - creating a new folder [955](#)
 - deleting files and folders from a profile [956](#)
- Profile Information page [976](#)
- profile manifest file [665](#)
- Profile Registry page [986](#)
- Profile Requirements page [978](#)
- Profile Shortcuts page [984](#)
 - creating a new profile shortcut [964](#)
 - excluding or deleting a profile shortcut [965](#)
 - including an existing shortcut [964](#)
- profiles [660](#), [664](#), [939](#)
- prog IDs [1386](#)
 - checking [1386](#)
- Prog IDs view [1422](#)
- programs (in Configuration Manager Web Console) [1512](#)
- Progress panel [377](#)
 - Merge Wizard [377](#)
- Project Information panel [378](#)
 - OS Snapshot Wizard [378](#)
- project path [465](#), [475](#), [483](#), [491](#), [497](#)
 - specifying in Repackaging Wizard [465](#), [475](#), [483](#), [491](#), [497](#)
- Project Properties dialog box [593](#), [594](#)
 - Exclusions tab [594](#)
 - General tab [593](#)
- ProjectName [145](#)
- projects [121](#), [122](#), [127](#), [128](#), [129](#), [131](#), [139](#), [153](#), [154](#)

Index

- creating [127](#)
- creating new [153](#)
- deleting [129](#)
- executing [128](#)
- filtering [128](#)
- integration with Workflow Manager workflows [139](#)
- New Workflow Project Wizard [153](#)
- running associated tools in [121](#), [128](#)
- saving [129](#)
- Source Package panel [154](#)
- using [127](#)
- Projects tab [139](#)
- ProjectTemplate [628](#)
- properties [152](#), [546](#), [1023](#), [1046](#), [1049](#), [1050](#), [1051](#)
 - adding custom setup [1050](#)
 - changing Add/Remove Programs [1049](#)
 - configuring tools in AdminStudio [152](#)
 - editing dialog [1046](#)
 - feature [1023](#)
 - modifying setup [1051](#)
 - removing custom setup [1051](#)
 - viewing Repackager project [546](#)
- Properties Dialog [1350](#)
- Properties dialog box [1064](#)
 - Microsoft patches [1350](#), [1351](#), [1352](#)
- Properties Tab
 - Tool Properties dialog [149](#)
- Properties tab [149](#)
- protocol
 - of App-V deployment server [824](#)
- pseudo-tokens [1257](#)
- Publication
 - subscribing [291](#)
 - subscribing to more than one [283](#)
- publication [284](#)
 - adding users [290](#)
 - assigning permissions to [290](#)
 - creating [286](#)
 - deleting [290](#)
 - editing [289](#)
 - editing an access list [290](#)
 - managing [284](#)
 - publishing [288](#)
 - setting a schedule [289](#)
- Publication Manager Dialog [355](#)
 - Application Catalog Replication [304](#)
 - using to manage publications [285](#)
 - using to manage subscriptions [291](#)
- Publication Processing dialog [356](#)
- Publication Properties dialog [356](#)
- Publication Wizard [380](#), [381](#), [382](#), [383](#)

- Publication Access List Panel [382](#)
- Publication Details Panel [381](#)
- Task Scheduling dialog box [359](#)
- Publisher URL [1086](#)
- publishing [281](#)
 - Application Catalog [281](#), [284](#)
 - database [284](#)
- PunchThrough Element [681](#)
- pv [516](#)
- PVKFile [1370](#)

Q

- Q [426](#), [437](#)
- QualityMonitor [1373](#), [1375](#), [1379](#), [1383](#), [1384](#), [1385](#), [1386](#), [1387](#), [1388](#), [1389](#), [1390](#), [1392](#), [1404](#), [1405](#), [1406](#), [1408](#), [1418](#)
 - checking class IDs [1384](#)
 - checking deployment status [1397](#)
 - checking file associations [1385](#)
 - checking help files [1385](#)
 - checking manifest files [1388](#)
 - checking ODBC data sources [1389](#)
 - checking ODBC drivers [1390](#)
 - checking prog IDs [1386](#)
 - checking services [1386](#)
 - checking shortcuts [1387](#)
 - checking type libraries [1387](#)
- Component Properties dialog [1409](#)
- creating exclusion lists [1390](#)
- deployment testing [1383](#)
- dialogs [1408](#)
 - excluding system files [1390](#)
- exclusion lists [1390](#)
- Feature Properties dialog [1410](#)
- Files dialog [1412](#)
- Install or Configure Feature Dialog [1411](#)
- Install or Configure Product Dialog [1411](#)
- install or configure products or features [1400](#)
- lockdown and runtime testing [1392](#)
- menus and toolbar [1406](#)
- MSI Doctor [1397](#), [1399](#), [1400](#), [1401](#), [1402](#)
- Product Properties dialog [1416](#)
- reinstall components [1402](#)
- Re-install Feature Dialog [1417](#)
- reinstall features [1401](#)
- Re-install Product Dialog [1417](#)
- running from the command line [1405](#)
- Test Cases [1375](#)
- test reports [1404](#)
- verifying files [1399](#)

- viewing deployment status properties [1399](#)
- viewing Test Item details [1379](#)
- views [1418](#)
- QualityMonitor projects [1374](#), [1375](#)
 - creating new [1374](#)
 - opening existing [1375](#)
- Quick Start evaluation experience [210](#)

R

- reboot handling [518](#)
- reference [133](#), [153](#), [378](#)
 - AdminStudio Interface [133](#)
 - for writing custom rules [681](#)
 - New Workflow Project Wizard [153](#)
 - OS Snapshot Wizard [378](#)
- REG files [1029](#)
 - importing [1029](#)
- registry [571](#), [572](#), [625](#), [1093](#)
 - exclusions [571](#), [625](#)
 - Import REG File Wizard [1093](#)
 - modifying in App-V Assistant [807](#), [808](#)
 - modifying in ThinApp Assistant [878](#)
 - removing exclusions [572](#)
- registry conflicts [1118](#)
- Registry Element [681](#)
- registry entries [1027](#)
- Registry Entries view [600](#)
- registry exclusions [571](#), [572](#)
 - editing existing [571](#)
 - removing global [572](#)
- registry information [1030](#)
 - removing [1030](#)
- Registry Isolation Options dialog box [848](#), [913](#), [996](#)
- registry keys [559](#), [1028](#)
 - and App-V application [808](#)
 - and Citrix profile [968](#)
 - and ThinApp application [879](#)
 - creating [1028](#)
 - excluding [559](#)
- Registry page [902](#)
- Registry Report [1132](#), [1134](#)
- Registry tab [625](#)
 - Exclusions Editor [625](#)
- registry values [560](#), [1029](#)
 - creating [1029](#)
 - excluding [560](#)
- Registry view [323](#), [327](#), [331](#), [1028](#), [1029](#), [1030](#), [1075](#)
 - creating a registry key in [1028](#)
 - creating a registry value in [1029](#)
 - importing REG files in [1029](#)
 - removing registry information from [1030](#)
- reimporting packages [1132](#)
- Remote Link Application Catalog panel [374](#)
- Removable Disk Changes go to Sandbox [897](#)
- Remove button [1050](#)
 - disabling in Add/Remove Programs [1050](#)
- RemoveFileData ACE [1176](#)
- RemoveIniFileData ACE [1176](#)
- RemoveRegistryData ACE [1177](#)
- reordering [125](#)
 - tasks [125](#)
- Repack.ini file [638](#)
- Repack.log [635](#)
- Repackaged Output view [532](#), [606](#), [611](#)
 - building InstallShield Editor projects [532](#)
 - building MSI packages [532](#)
 - creating virtual applications [646](#)
- Repackager [443](#), [449](#), [454](#), [516](#), [523](#), [524](#), [525](#), [526](#), [527](#), [531](#), [541](#), [546](#), [547](#), [548](#), [549](#), [550](#), [558](#), [559](#), [560](#), [561](#), [562](#), [563](#), [564](#), [582](#), [584](#), [586](#), [587](#), [591](#), [592](#), [593](#), [595](#), [596](#), [599](#), [600](#), [602](#), [603](#), [604](#), [605](#), [606](#), [611](#), [612](#), [614](#), [619](#), [621](#), [628](#), [639](#), [640](#)
 - About Repackager dialog [587](#)
 - additional setup programs [463](#)
 - Advanced Settings view [612](#)
 - and anti-virus software [451](#)
 - automatically building a Citrix profile [543](#)
 - automatically building a ThinApp application [543](#)
 - automatically creating a Windows Installer package after creating the Editor project [543](#)
 - automatically generating a virtual application during project build [650](#)
 - automatically running PackageExpert tests upon build [538](#), [609](#)
 - building a Citrix profile [532](#), [610](#), [662](#), [668](#), [695](#)
 - building a ThinApp application [695](#)
 - building a ThinApp applications [659](#)
 - building a VMware ThinApp application [610](#)
 - building an App-V application [695](#)
 - building an App-V package [609](#)
 - building Citrix profile automatically at project build [543](#)
 - building InstallShield Editor projects [532](#)
 - building MSI packages [532](#)
 - building ThinApp application automatically at project build [543](#)
- Captured Installation view [596](#)
- changing data type appearance [524](#)
- clean systems [449](#)
- command-line options [516](#)
- component settings options [542](#), [613](#)
- configuring advanced conversion options [541](#)

- configuring exclusions [558](#)
- conversion problems [639](#), [640](#)
- converting .axt [527](#)
- converting .inc [527](#)
- converting .ipf [527](#)
- converting .isl [531](#)
- converting .txt [531](#)
- converting .wse [531](#)
- converting InstallShield log files [531](#)
- converting legacy setups [526](#)
- converting Novell ZENworks projects [527](#)
- converting Repackager 3.x output [527](#)
- converting SMS projects [527](#)
- converting WinINSTALL projects [531](#)
- converting Wise Installation projects [531](#)
- Create Report dialog box [587](#)
- creating projects [525](#)
- creating reports [548](#)
- creating virtual applications [643](#)
- Deleted Files view [604](#)
- Deleted Registry Entries view [605](#)
- detecting dependencies [547](#)
- dialog boxes [586](#)
- editing generated projects in InstallShield Editor [550](#)
- Error Building Table File error [640](#)
- excluding all files in a directory [559](#)
- excluding all shortcuts in a directory [561](#)
- excluding directories and subdirectories [559](#)
- excluding files [558](#)
- excluding INI file sections [560](#)
- excluding INI files [560](#)
- excluding registry keys [559](#)
- excluding registry values [560](#)
- excluding shortcuts [561](#)
- excluding shortcuts from subdirectories [561](#)
- exclusion list [635](#)
- exclusions [564](#)
- file exclusions [558](#)
- Files and Folders view [599](#)
- files associated with [635](#)
- INI Files view [603](#)
- installing on a clean machine [457](#)
- installing on clean machine [457](#)
- InstallShield SmartScan options [542](#), [614](#)
- ISDEV fatal error -5023 [640](#)
- isolating applications [1355](#)
- Isolation Options dialog box [588](#)
- launching [523](#)
- launching remotely [454](#)
- location of isrepackager.ini [114](#)
- menus [584](#)
- modifying external configuration file [563](#)
- Options dialog box [591](#)
- Options.ini file [628](#)
- OtherComponentFileExtensions [628](#)
- package conversion options [541](#), [612](#)
- Package Information view [611](#)
- Password Required dialog box [592](#)
- Project Properties dialog box [593](#)
- Registry Entries view [600](#)
- remote repackaging on Windows NT4 [454](#)
- Repackaged Output view [606](#), [646](#)
- repackaging legacy setups [458](#)
- running automated tests against the .msi package at build [538](#), [609](#)
- running Windows Installer Converter from the command line [668](#)
- saving projects [549](#)
- scanning InstallShield Professional setups [573](#)
- setting digital signature options [588](#)
- setting manifest options [588](#)
- Setup Intent Wizard [619](#)
- Shortcuts view [602](#)
- SmartScan Wizard [614](#), [617](#)
- SMS conversion problems [640](#)
- specifying external configuration file [562](#)
- support of user-defined extensions in options.ini [628](#)
- ThinApp applications [659](#)
- toolbar [584](#)
- troubleshooting [639](#), [640](#)
- viewing properties [546](#)
- views [596](#)
- virtualization [609](#), [610](#)
- VMware Repackaging Wizard [621](#)
- Windows Installer Converter [643](#), [645](#), [668](#), [695](#)
- WinINSTALL Conversion dialog box [595](#)
- WinINSTALL conversion problems [639](#)
- Repackager 3.x output [527](#)
 - converting to Repackager project [527](#)
- Repackager error building table file [640](#)
- Repackager projects [525](#), [549](#)
 - creating [525](#)
 - saving [549](#)
- Repackager Start Page [582](#)
- repackaging [443](#), [450](#), [454](#), [458](#), [563](#)
 - a Windows Installer package [494](#)
 - alternate-language [450](#)
 - an installation from a self-extracting .exe [461](#)
 - and anti-virus software [572](#)
 - excluding processes from [514](#)
 - exclusions [563](#)
 - legacy setups [443](#), [458](#)

- remotely on Windows NT4 [454](#)
- support for 64-bit applications [446](#)
- Repackaging an InstallScript MSI Setup to a Basic MSI Setup [492](#)
- repackaging methods [450](#), [452](#)
 - selecting [469](#)
- Repackaging panel [510](#)
- Repackaging Using the Installation Monitoring Method [459](#)
- Repackaging Using the InstallScript Professional Logging Method [488](#)
- Repackaging Using the Snapshot Method [469](#)
- Repackaging Wizard [454](#), [493](#), [500](#), [501](#), [503](#), [504](#), [506](#), [508](#), [509](#), [510](#), [512](#), [513](#), [514](#), [515](#), [518](#)
 - additional panels [513](#)
 - additional setup programs [463](#)
 - Additional Setup Programs dialog box [513](#)
 - Analysis Options dialog box [515](#)
 - Collect Product Information panel [504](#)
 - deleting added setups [464](#)
 - Excluded Processes dialog box [514](#)
 - InstallScript MSI Conversion Output panel [509](#)
 - InstallScript MSI Identified panel [506](#)
 - InstallShield Professional Setup panel [508](#)
 - Method Selection panel [501](#)
 - Original InstallShield Professional Setup panel [616](#)
 - reboot handling [518](#)
 - repackaging a Windows Installer package [494](#)
 - Repackaging panel [510](#)
 - running from command line [493](#)
 - running remotely on Windows NT4 [454](#)
 - Set Target Project Information and Capture Settings panel [508](#)
 - Setup Information dialog box [514](#)
 - Snapshot Method panel [503](#)
 - specifying project path [465](#), [475](#), [483](#), [491](#), [497](#)
 - Summary panel [512](#)
 - using [459](#)
 - Welcome panel [500](#)
- Repackaging Wizard Command-Line Options [516](#)
- Repair button [1050](#)
 - disabling in Add/Remove Programs [1050](#)
- Replication
 - configuring servers for [277](#)
 - excluded tables [404](#), [405](#)
- replication [276](#), [296](#)
 - Application Catalog [276](#), [277](#), [281](#)
 - configuring servers for Replication [277](#)
 - deleting a subscription [295](#)
 - enabling/disabling a subscription [294](#)
 - lifecycle [281](#)
 - managing publications [285](#)
 - publishing a publication [288](#)
 - publishing Publication manually [283](#)
 - setting a publication schedule [289](#)
 - subscribing to more than one Publication [283](#)
 - tables included and excluded [404](#)
 - updating a subscription manually [283](#), [295](#)
- Replication Command Line Options dialog [426](#)
- Report Center [1695](#), [1722](#)
 - available reports [1704](#)
 - evaluating [73](#)
 - Package reports [1700](#)
- Report View [1722](#)
- reports [548](#), [1132](#), [1135](#), [1136](#), [1583](#), [1584](#), [1585](#), [1588](#), [1593](#), [1722](#)
 - All Reports page [1722](#)
 - assigning Role permission to view [1734](#)
 - ConflictSolver [1132](#), [1136](#)
 - creating [1722](#)
 - creating custom [1136](#)
 - creating in Repackager [548](#)
 - Crystal Reports [1132](#), [1135](#)
 - editing Role permissions to view [1734](#)
 - File and Registry [1134](#)
 - Files [1132](#)
 - Job Details Report (Predeployment Test) [1588](#)
 - Job Summary Report (Predeployment Test) [1585](#)
 - Machine Result Summary Report (Predeployment Test) [1593](#)
 - Package [1132](#), [1134](#)
 - Predeployment Test Results By Job Report (Predeployment Test) [1584](#)
 - Predeployment Test Results By Machine Report (Predeployment Test) [1593](#)
 - Predeployment Test) [1583](#)
 - Registry [1132](#)
 - Report Center [1695](#), [1722](#)
 - Report View [1722](#)
 - Report Wizard [1733](#)
 - Test Details Report (Predeployment Test) [1589](#)
 - Test Result Summary Report (Predeployment Test) [1586](#)
 - viewing Predeployment Test result reports [1564](#)
- Reports Wizard [1733](#)
- requirements
 - how they are applied for a Citrix profile [949](#)
 - setting language requirements in Citrix profile [948](#)
- Reset Sandbox on Exit [897](#)
- Resolution Details dialog [1247](#)
- Resolution Operations dialog [1232](#), [1247](#)
- resolution options [1121](#), [1243](#)
 - changing ConflictSolver [1121](#)
 - ConflictSolver [1243](#)

Resolution Options dialog [1248](#)
 Resolution tab [1243](#)
 in ConflictSolver Options dialog [1243](#)
 resolving
 a manual test in PackageExpert [1298](#)
 errors in PackageExpert [1293](#)
 resolving conflicts [1127](#)
 response transforms [1016](#)
 restricted environments [1395](#)
 running lockdown and runtime tests [1395](#)
 result filters
 deleting in PackageExpert [1290](#), [1313](#)
 Results panel [620](#)
 Setup Intent Wizard [620](#)
 Role [197](#), [199](#)
 assigning permission to view reports [1734](#)
 creating [197](#)
 Role Administration Page [199](#)
 Role Details View [201](#)
 roles
 copying [198](#)
 creating [197](#)
 deleting [198](#)
 system roles [194](#)
 user roles [194](#)
 root folder name [822](#)
 RTSP [789](#), [824](#)
 RTSPS [789](#), [824](#)
 Rule data binding [690](#)
 Rule Element [681](#)
 rules
 for pre-conversion evaluation [677](#)
 Rules Viewer dialog [1248](#)
 Rules Wizard [1253](#), [1254](#), [1256](#), [1257](#), [1258](#)
 Additional Information panel [1254](#), [1256](#)
 Custom ACEs panel [1182](#), [1184](#)
 DLL-Based ACEs panel [1258](#)
 General Information panel [1254](#)
 inserting Column Names in Error or Display Strings [1257](#)
 inserting Product Name in Error or Display strings [1257](#)
 pseudo-tokens [1257](#)
 Summary panel [1258](#)
 Token Grammar [1257](#)
 User-Defined ACEs [1182](#), [1184](#)
 Welcome panel [1253](#)
 Where Clause Panel [1257](#)
 running associated tools [121](#), [128](#)
 runtime testing [1392](#)

S

-S [426](#)
 sample.mdb [216](#)
 sandbox cache [858](#), [859](#)
 Sandbox Name [896](#)
 saving [129](#)
 projects [129](#)
 workflows [129](#)
 -sb [516](#)
 Scan for Dependencies [1218](#)
 ConflictSolver [1218](#)
 scanning for dependencies [1108](#)
 option disabled [1108](#)
 packages in software repository [1108](#)
 Scanning Media panel [618](#)
 SmartScan Wizard [618](#)
 Scanning panel [617](#)
 Scanning project panel [620](#)
 Setup Intent Wizard [620](#)
 Schedule panel [376](#)
 Package Auto Import Wizard [376](#)
 schema
 AdminStudio database schema tables [386](#)
 Script Execution dialog box [991](#)
 script-based setup.exe [1461](#)
 creating with Distribution Wizard [1461](#)
 Search Results Tab [316](#), [1208](#)
 Security Console
 evaluating [73](#)
 Security tables [402](#)
 Select a Directory panel [701](#)
 Select a Package dialog box [1321](#)
 Select a Suite of Packages panel [702](#)
 Select Conversion Target panel [697](#)
 Select Windows Installer Package panel [700](#)
 Windows Installer Converter [700](#)
 Sequencer [783](#)
 Sequencer. See also Microsoft App-V Sequencer.
 sequences [1045](#)
 restoring dialog [1045](#)
 server component
 FlexWrap [1637](#)
 server locations [1048](#), [1049](#)
 adding [1048](#)
 modifying [1048](#)
 removing [1048](#)
 reordering [1049](#)
 Server Locations view [1048](#), [1049](#), [1085](#)
 adding server locations in [1048](#)
 modifying server locations in [1048](#)

- removing server locations in [1048](#)
- reordering server locations in [1049](#)
- Service Packs Requirement dialog box [998](#)
- service type [1043](#)
 - setting [1043](#)
- ServiceControlEvents [628](#)
- services [1386](#)
 - and Citrix profiles [761](#)
 - and Citrix XenApp [762](#)
 - and Windows Installer Converter [761](#)
 - checking [1386](#)
- Services view [1424](#)
- Set Target Project Information and Capture Settings panel [508](#)
 - Repackaging Wizard [508](#)
- Setup Feature Tree panel [617](#)
- Setup Information dialog box [514](#)
 - Repackaging Wizard [514](#)
- Setup Intent Wizard [547](#), [619](#), [620](#)
 - Results panel [620](#)
 - Scanning project panel [620](#)
 - Welcome panel [619](#)
- Setup Organization [1020](#)
- setup programs
 - adding additional in Repackaging Wizard [463](#)
- setup properties [1050](#), [1051](#)
 - adding custom [1050](#)
 - customizing [1050](#)
 - modifying [1051](#)
 - removing custom [1051](#)
- Setup Properties view [1050](#), [1051](#), [1086](#)
 - adding and editing comments [1051](#)
 - adding custom setup properties [1050](#)
 - modifying properties [1051](#)
 - removing custom setup properties from [1051](#)
- setup property comments [1051](#)
 - adding and editing [1051](#)
- Setup.exe [1055](#)
 - creating for package and transform [1055](#)
- Setup.exe panel [1095](#)
 - Packaging Wizard [1095](#)
- Setup.ini [1056](#)
- setups [443](#), [1045](#)
 - deleting added in Repackaging Wizard [464](#)
 - disabling custom [1045](#)
 - repackaging legacy [443](#), [458](#)
- severity
 - changing for a PackageExpert message [1282](#), [1283](#)
- SFT file [656](#)
- SFT package
 - specifying root folder name [787](#)

- SFT_SOFTGRIDSERVER [789](#)
- shared assemblies [1363](#)
 - servicing published [1363](#)
- shared location [114](#)
- SharedCommonFiles [628](#)
- SharedPoint [145](#)
- shortcut [438](#)
 - creating to a specific Application Catalog [438](#)
- shortcut conflicts [1118](#)
- Shortcut Element [681](#)
- shortcuts [561](#), [1030](#), [1031](#), [1032](#), [1033](#), [1077](#), [1079](#), [1080](#), [1387](#)
 - changing icon for [1031](#)
 - changing location for [1032](#)
 - changing target for [1032](#)
 - checking [1387](#)
 - creating [1031](#)
 - creating a hot key for [1032](#)
 - determining path of changed in Tuner [1033](#)
 - excluding [561](#)
 - excluding all in a directory [561](#)
 - excluding from subdirectories [561](#)
 - excluding or deleting from App-V application [805](#)
 - excluding or deleting from Citrix profile [965](#)
 - excluding or deleting from ThinApp application [876](#)
 - including in App-V application [805](#)
 - including in Citrix profile [964](#)
 - including in ThinApp application [876](#)
 - location [1080](#)
 - properties [1077](#)
 - removing [1033](#)
 - target [1079](#)
 - when to exclude or delete from App-V application [807](#)
 - when to exclude or delete from Citrix profile [966](#)
 - when to exclude or delete from ThinApp application [878](#)
- Shortcuts view [324](#), [332](#), [602](#), [1031](#), [1032](#), [1033](#), [1076](#), [1422](#)
 - changing a shortcut's icon [1031](#)
 - changing a shortcut's location [1032](#)
 - changing a shortcut's target in [1032](#)
 - creating a hot key for a shortcut [1032](#)
 - creating shortcuts in [1031](#)
 - OS Snapshot [328](#)
 - removing shortcuts from [1033](#)
- silent activation [73](#)
- silent mode [437](#)
- Single Application option [699](#)
- SISAuthor=Repackager [628](#)
- SmartScan Wizard [573](#), [614](#), [615](#), [618](#)
 - automatic launch [614](#)
 - conditions for automatic launch [614](#)
 - launching automatically [614](#)

- option to match legacy system's OS requirements [612](#)
- Repackager [617](#)
- Scanning Media panel [618](#)
- Welcome panel [615](#)
- SMS [1054](#), [1444](#)
 - creating file [1054](#)
 - deploying [1055](#)
 - using Distribution Wizard to prepare packages for [1444](#)
- SMS conversion problems [640](#)
 - Repackager [640](#)
- SMS distribution [1445](#)
- SMS Distribution Panel [1472](#)
 - Distribution Wizard [1472](#)
- SMS panel [1095](#)
 - Packaging Wizard [1095](#)
- SMS project [527](#)
 - converting to Repackager project [527](#)
- SMS templates [1445](#)
- sn [516](#)
- snapshot [450](#), [452](#)
- Snapshot method
 - and anti-virus software [451](#)
- Snapshot Method panel [503](#)
 - Repackaging Wizard [503](#)
- sNotification [424](#)
- software publishing credentials [1359](#)
- Software Repository [250](#)
 - and ConflictSolver [258](#)
 - and Distribution Wizard [258](#)
 - and InstallShield Editor [258](#)
 - and Predeployment Test Preparation Wizard [258](#)
 - conflict resolution and [1758](#)
 - enabling in Application Catalogs [251](#)
 - Get Latest Version [258](#)
 - getting a copy of a package [258](#)
 - importing packages into [253](#)
 - introduction [250](#)
 - package check in and check out [257](#)
 - using [250](#)
- software repository
 - scanning for dependencies for files stored in Software Repository [1108](#)
- Source Application Catalog panel [377](#)
 - Merge Wizard [377](#)
- Source Package panel [154](#), [1250](#)
 - Conflict Wizard [1250](#)
 - in New Workflow Project Wizard [154](#)
- Source Type panel [1250](#)
 - Conflict Wizard [1250](#)
- SourcePackage [145](#)
- sp_addpublication [283](#)

- SPCFile [1370](#)
- SPD File Details [1474](#)
 - Distribution Wizard [1474](#)
- SPD Parameters panel [1473](#)
 - Distribution Wizard [1473](#)
- Specifying duplicate package identifiers in Application Manager [234](#)
- SQL Express [210](#)
- SQL Server
 - merge replication [296](#)
 - preferred authentication mode [268](#)
- SQL Server Enterprise Manager [283](#)
 - using to create Publications and Subscriptions [283](#)
- standalone Application Catalog [209](#)
- Standard.nir [635](#)
- start name [1043](#)
 - setting NT service [1043](#)
- Start Page [134](#)
- start type [1043](#)
 - setting NT service [1043](#)
- status [1522](#)
 - viewing Job Manager Engine status [1780](#)
 - viewing status of distributed package in Configuration Manager Web Console [1522](#)
- stblConflictConfiguration [410](#)
- stblConflictTransformList [414](#)
- stblIgnoreTableList [418](#)
- subscribing [291](#)
- subscribing to [276](#)
 - Application Catalog [276](#), [277](#), [281](#), [291](#)
 - Publication [291](#)
- Subscription
 - disabling [283](#)
 - enabling [283](#)
- subscription [266](#)
 - creating [292](#)
 - deleting [295](#)
 - enabling or disabling [294](#)
 - in Application Manager [266](#)
 - manually updating [295](#)
- Subscription Manager [291](#)
- Subscription Manager dialog [357](#)
- Subscription Wizard [383](#), [384](#), [385](#)
- Subscriptions dialog [351](#)
- suiting
 - in App-V Assistant [810](#)
- Summary panel [373](#), [376](#), [512](#), [1253](#), [1258](#), [1259](#), [1364](#)
 - Application Isolation Wizard [1364](#)
 - Conflict Wizard [1253](#)
 - Import Wizard [373](#)
 - Package Auto Import Wizard [376](#)

- Repackaging Wizard [512](#)
- Rules Wizard [1258](#)
- Validation Wizard [1259](#)
- suppression issues [1047](#)
- SuppressReboot [1056](#)
- SuppressWin2k [1056](#)
- System Configuration view [1074](#)
 - Tuner [1074](#)
- system roles [194](#)
- System tables [391](#)
- system tray mode [437](#)

T

- table [1057](#)
 - adding a new record using Direct Editor [1057](#)
 - adding a new row using Direct Editor [1057](#)
- table names [391](#)
 - Application Catalog [391](#)
- tables [404](#)
 - excluded from Replication [404](#), [405](#)
 - in Application Catalog Replication [391](#)
 - included in replication [404](#)
- Tables view [325](#), [328](#), [332](#), [1219](#), [1220](#)
- taking OS snapshots [302](#)
- target collection [1499](#)
- target directory [154](#)
 - setting in New Workflow Project Wizard [154](#)
- Target Directory and File Name panel [154](#)
 - in New Workflow Project Wizard [154](#)
- Target Information panel [1252](#)
 - Conflict Wizard [1252](#)
- TargetDir [145](#), [154](#)
- TargetFileName [145](#), [154](#)
- task page help [115](#)
 - setting location in AdminStudio [115](#)
- task properties [124](#)
 - modifying [124](#)
- tasks [120](#), [124](#), [125](#), [126](#), [138](#)
 - associating help files [126](#)
 - associating tools with in AdminStudio [120](#)
 - creating new [124](#)
 - creating notes for [125](#)
 - deleting [126](#)
 - in workflows [138](#)
 - renaming [125](#)
 - reordering in Workflow Templates tab [125](#)
- template.pdf [1445](#)
- template.sms [1445](#)
- Test & Resolve view [1309](#)
- Test Cases [1375](#), [1379](#), [1380](#), [1381](#), [1382](#), [1402](#)
 - adding comments [1379](#)
 - clearing results [1380](#)
 - creating custom [1402](#)
 - filtering data [1382](#)
 - manually setting status [1381](#)
- Test Cycle Summary view [1420](#)
- Test Details Report (Predeployment Test) [1589](#)
- Test Item details [1379](#)
 - viewing [1379](#)
- Test Item Information dialog [1418](#)
- Test Items [1376](#), [1377](#), [1378](#), [1382](#)
 - adding comments [1378](#)
 - manually setting status [1382](#)
 - running individual [1376](#)
 - running multiple [1377](#)
- Test Packages panel [704](#)
- Test Progress dialog [1418](#)
- test reports [1404](#)
- Test Result dialog [1418](#)
- Test Result Filter Editor dialog box [1322](#)
- test result message
 - adding [1292](#)
 - deleting in PackageExpert [1293](#)
 - editing in PackageExpert [1292](#)
- Test Result Summary Report (Predeployment Test) [1586](#)
- test results
 - adding ad-hoc test results [1290](#)
 - customizing in PackageExpert [1282](#)
 - deleting [1281](#)
 - deleting a test result message [1293](#)
 - deleting catalog [1282](#)
 - deleting local [1281](#)
 - editing in PackageExpert [1292](#)
 - viewing and managing [1271](#)
 - viewing Catalog results [1279](#)
 - viewing in ConflictSolver Test Results view [1280](#)
 - viewing in PackageExpert [1272](#), [1273](#)
 - viewing Local results [1279](#)
- Test Results view [1217](#), [1269](#)
- testing [1383](#), [1392](#)
 - deployment [1383](#)
 - in PackageExpert [1266](#)
 - lockdown and runtime [1392](#)
 - Predeployment Test [1551](#), [1552](#)
- test-ready package [1558](#)
 - installing [1562](#)
- ThinApp [516](#), [543](#), [659](#), [662](#)
 - AppLink [915](#)
 - command line [516](#)
 - Interm directory [661](#)
 - Virtual Operating System [660](#)

- ThinApp application [643](#)
- ThinApp applications [543](#), [645](#), [659](#), [695](#)
 - about [659](#), [851](#), [855](#)
 - about virtual suites [649](#)
 - adding an AppLink reference [886](#)
 - adding diagnostic tools to [861](#), [910](#)
 - adding existing folder [865](#)
 - adding files to [864](#)
 - adding or deleting registry keys and values [879](#)
 - and Active Directory [860](#)
 - Application Link [885](#)
 - AppLink [885](#), [915](#)
 - AppSync [888](#)
 - automatically creating from Repackager [646](#)
 - benefits of deploying [662](#), [852](#)
 - building [891](#)
 - building a Windows Installer package with build output [883](#)
 - building using command line [925](#)
 - components of [660](#), [855](#)
 - compressing [874](#)
 - configuration file (package.ini) [925](#)
 - controlling access to [860](#), [861](#)
 - conversion error and warning messages [925](#)
 - creating using InstallShield ThinApp Assistant [645](#)
 - creating using Repackager [668](#)
 - creating using Windows Installer Converter [668](#)
 - creating with InstallShield [850](#)
 - defining shortcut executables [874](#)
 - excluding vs. deleting a shortcut [878](#)
 - files included in [660](#), [855](#)
 - how transforms are included [857](#)
 - including an existing shortcut [876](#)
 - inheritance of isolation options from folders to files [872](#)
 - inheritance of isolation options in the registry [880](#)
 - Interm directory [661](#), [857](#)
 - intermediate data files [661](#), [857](#)
 - launching [855](#)
 - linking ThinApp applications [885](#), [915](#)
 - linking to an application with more than one shortcut [919](#)
 - linking to an application with one shortcut [919](#)
 - linking to another ThinApp application [919](#)
 - location of [856](#)
 - managing files and folders [863](#)
 - methods to convert Windows Installer packages [644](#)
 - modifying registry [878](#)
 - moving files and folders [867](#)
 - operating system [851](#)
 - order of import of linked ThinApp applications [917](#)
 - overview of isolation options [869](#)
 - overview of ThinApp Assistant [851](#)
 - Package.DAT [660](#)
 - Package.DAT file [855](#)
 - prerequisites for building [662](#), [861](#)
 - relative paths when defining AppLink references [886](#)
 - renaming a shortcut [878](#)
 - sandbox cache [859](#)
 - sandbox name [859](#)
 - sandboxes [858](#)
 - security and authorization for linked applications [918](#)
 - selecting application shortcuts [872](#)
 - Setting AppLink options [885](#)
 - setting AppSync Expiration settings [889](#)
 - setting folder isolation options [912](#)
 - setting isolation options [869](#), [870](#)
 - setting isolation options for folders and files [871](#)
 - setting Log Monitor tracing options [883](#)
 - setting registry isolation options [879](#)
 - shortcut requirements [875](#), [901](#)
 - shortcuts [660](#)
 - steps to create [858](#)
 - steps to create with ThinApp Assistant [853](#)
 - testing prior to conversion [673](#)
 - ThinApp Assistant [850](#)
 - ThinAppPackage directory [856](#)
 - updating applications [888](#)
 - updating using AppSync [921](#)
 - virtual environment [851](#)
 - virtual operating system [660](#)
 - Virtual Operating System (VOS) [851](#)
 - virtualization [695](#)
 - Windows Installer Converter [695](#)
- ThinApp Assistant [544](#), [645](#), [850](#), [925](#)
 - about [852](#)
 - about sandboxes [858](#)
 - adding an existing folder to a ThinApp application [865](#)
 - adding diagnostic tools to a ThinApp application [861](#), [910](#)
 - adding files to a ThinApp application [864](#)
 - adding or deleting registry keys and values from a ThinApp application [879](#)
 - application features requiring pre- or post-conversion actions [925](#)
 - Applications page [901](#)
 - Build Options page [904](#)
 - building a ThinApp application [891](#)
 - building a Windows Installer package with build output [883](#)
 - building ThinApp application in Direct Edit mode [882](#)
 - clearing the cache [910](#)
 - Compression Type option [874](#)
 - compression types [874](#)

- controlling access to ThinApp applications [861](#)
- controlling the display of predefined folders [867](#)
- creating a ThinApp application [858](#)
- creating new application shortcut executables [875](#)
- creating new folder [866](#)
- default isolation options [871](#)
- deleting files and folders [867](#)
- Diagnostic Tools dialog box [861](#), [910](#)
- disabling Log Monitor Tracing [885](#)
- enabling ThinApp application building when editing a Windows Installer package [882](#)
- error messages [925](#)
- excluding or deleting a ThinApp application shortcut [876](#)
- Files & Folders page [898](#)
- Home page [894](#)
- how transforms are included [857](#)
- including an existing shortcut [876](#)
- inheritance of isolation options [872](#)
- inheritance of isolation options in registry [880](#)
- Interm directory [857](#)
- isolation options [870](#)
- managing files and folders [863](#)
- modifying build options [881](#)
- modifying registry settings [878](#)
- modifying shortcuts [872](#)
- moving files and folders [867](#)
- overview [851](#)
- overview of [851](#)
- overview of isolation options [869](#)
- Package Information page [896](#)
- package.ini [925](#)
- prerequisites for creating ThinApp application [861](#)
- reference [893](#)
- Registry Isolation Options dialog box [913](#)
- Registry page [902](#)
- renaming a shortcut [878](#)
- sandbox cache [859](#)
- selecting releases to build [881](#)
- setting isolation options [869](#), [870](#)
- setting isolation options for folders and files [871](#)
- setting isolation options on folders [912](#)
- setting Log Monitor tracing options [883](#)
- setting registry isolation options [879](#)
- specifying access via Active Directory [860](#)
- specifying general settings [859](#)
- steps to create a ThinApp application [853](#)
- supported InstallShield project types [857](#)
- when to exclude and when to delete shortcuts [878](#)
- Write Copy, Merged, and Full isolation options [870](#)
- ThinApp cache
 - clearing [910](#)

- ThinApp Log Monitor [883](#)
- ThinApp Virtual Operating System (VOS) [851](#)
- ThinApp Virtualization Suite Not Found [925](#)
- ThinApp VOS [660](#)
- ThinAppPackage directory [856](#)
- TimeStampAssemblies [1370](#)
- Tivoli [1446](#)
 - using Distribution Wizard to prepare packages for [1446](#)
- Tivoli Integration panel [1472](#)
 - Distribution Wizard [1472](#)
- Tivoli Settings panel [1472](#)
 - Distribution Wizard [1472](#)
- Token Grammar [1257](#)
 - Rules Wizard [1257](#)
- tool accessibility [122](#)
- Tool Properites panel [152](#)
- tool properties [149](#)
 - viewing [149](#)
- Tool Properties dialog [149](#), [150](#)
 - Configuration tab [150](#)
 - Properties tab [149](#)
- Tool Properties panel [152](#)
 - Add Tool Wizard [152](#)
- toolbar [140](#), [305](#), [584](#), [1059](#), [1200](#), [1406](#)
 - AdminStudio [140](#)
 - Application Manager [305](#)
 - ConflictSolver [1200](#)
 - QualityMonitor [1406](#)
 - Repackager [584](#)
- Toolbars tab [1064](#), [1235](#)
- tools [117](#), [118](#), [119](#), [120](#), [121](#), [126](#), [128](#), [137](#), [149](#), [152](#)
 - adding [118](#)
 - adding command-line configurations [119](#)
 - adding from Workflow Templates tab [126](#)
 - adding in AdminStudio [152](#)
 - adding using the Add Tool Wizard [152](#)
 - associating with tasks in AdminStudio [120](#)
 - deleting command-line configurations from in AdminStudio [120](#)
 - editing properties for existing [118](#)
 - modifying command-line configurations [119](#)
 - running associated in projects [121](#), [128](#)
 - viewing properties [149](#)
- Tools Gallery [121](#)
 - deleting tools from [121](#)
 - on Tools tab [137](#)
- Tools tab [137](#)
- transform [435](#), [1019](#), [1055](#)
 - applying to package during command-line import [435](#)
 - creating Setup.exe for [1055](#)
 - postvalidating [1019](#)

- transform files
 - associating with application catalog [230](#)
 - associating with package using ad-hoc import [250](#)
 - performing ad-hoc import [230](#), [236](#), [250](#)
- transform properties [1016](#)
 - viewing [1016](#)
- Transform Summary dialog [1064](#)
- transforms [1012](#), [1013](#), [1014](#), [1015](#), [1016](#), [1064](#), [1131](#)
 - creating generic [1015](#), [1064](#)
 - creating new [1013](#)
 - creating universal [1015](#), [1064](#)
 - generic [1015](#), [1064](#)
 - importing [238](#)
 - opening existing [1014](#)
 - opening recently accessed [1015](#)
 - resolving conflicts using [1131](#)
 - response [1016](#)
 - Tuner [1012](#)
 - universal [1015](#), [1064](#)
 - working with [1012](#)
- Tuner [1011](#), [1012](#), [1016](#), [1027](#), [1044](#), [1052](#), [1056](#), [1057](#), [1090](#)
 - adding and editing setup property comments [1051](#)
 - adding custom setup properties [1050](#)
 - adding files [1025](#)
 - adding INI files [1034](#)
 - adding new data sources [1038](#)
 - adding new INI file keys [1035](#)
 - adding new ODBC data source attributes [1038](#)
 - adding new ODBC driver attributes [1038](#)
 - adding sections to INI files [1035](#)
 - adding server locations [1048](#)
 - CAB files [1027](#)
 - changing a feature's visibility [1021](#)
 - changing a shortcut's icon [1031](#)
 - changing a shortcut's location [1032](#)
 - changing a shortcut's target [1032](#)
 - changing Add/Remove Programs Properties [1049](#)
 - creating a hot key [1032](#)
 - creating a new transform file [1068](#)
 - creating a registry key [1028](#)
 - creating a registry value [1029](#)
 - creating shortcuts [1031](#)
 - Direct Editor [1057](#), [1090](#)
 - disabling custom setups [1045](#)
 - displaying files from the base Windows Installer package [1026](#)
 - editing dialog properties [1046](#)
 - editing ODBC data source attributes [1039](#)
 - editing ODBC driver attributes [1039](#)
 - hiding dialogs during UI sequences [1044](#)
 - importing existing INI files [1034](#)
 - importing REG files [1029](#)
 - modifying server locations [1048](#)
 - modifying setup properties [1051](#)
 - opening a recent transform file [1070](#)
 - opening an existing transform file [1071](#)
 - preparing packages for distribution [1052](#)
 - preventing installation of files from the Windows Installer package [1026](#)
 - removing added files [1027](#)
 - removing custom setup properties [1051](#)
 - removing existing ODBC data sources [1039](#)
 - removing INI file keys [1037](#)
 - removing INI files [1036](#)
 - removing ODBC data source attributes [1040](#)
 - removing registry information [1030](#)
 - removing sections from INI files [1037](#)
 - removing server locations [1048](#)
 - removing shortcuts [1033](#)
 - reordering server locations [1049](#)
 - restoring dialog sequences [1045](#)
 - setting initial state of a feature [1021](#)
 - setting the NT service arguments [1040](#)
 - setting the NT service dependencies [1041](#)
 - setting the NT service description [1041](#)
 - setting the NT service display name [1041](#)
 - setting the NT service error control level [1042](#)
 - setting the NT service load order group [1042](#)
 - setting the NT service overall install result [1042](#)
 - setting the NT service start name and password [1043](#)
 - setting the NT service start type [1043](#)
 - setting the NT service type [1043](#)
 - Setup.ini [1056](#)
 - suppressing License Agreement dialog [1045](#)
 - transforms [1012](#)
 - validation in [1016](#)
 - working with dialogs [1044](#)
- Tuner reference [1058](#), [1059](#), [1062](#), [1063](#), [1064](#), [1067](#), [1068](#), [1070](#), [1071](#)
 - checklist [1062](#)
 - customization steps [1062](#)
 - Customize dialog box [1064](#)
 - Help view [1071](#)
 - InstallShield Start page [1068](#)
 - menus [1059](#)
 - Options dialog box [1064](#)
 - Output Window [1063](#)
 - Package Validation view [1071](#)
 - Properties dialog box [1064](#)
 - toolbar [1059](#)
 - user interface reference [1058](#)

- View Bar [1062](#)
- views [1067](#)
- Tuner views [1067](#), [1068](#), [1071](#), [1072](#), [1073](#), [1074](#), [1075](#), [1076](#), [1081](#), [1082](#), [1083](#), [1085](#), [1086](#), [1087](#), [1088](#), [1089](#), [1090](#)
- Add/Remove Programs [1086](#)
- Additional Tools [1090](#)
- Application Configuration [1085](#)
- Dialogs [1086](#)
- Direct Editor [1090](#)
- Features [1073](#)
- Files and Folders [1075](#)
- Help [1071](#)
- INI Files [1081](#)
- Location [1090](#)
- NT Services [1083](#)
- ODBC Resources [1082](#)
- Package [1089](#)
- Package Preparation [1087](#)
- Package Validation [1071](#)
- Postvalidation [1088](#)
- Prevalidation [1071](#)
- Product Properties [1072](#)
- Registry [1075](#)
- Server Locations [1085](#)
- Setup Properties [1086](#)
- Setup.exe [1090](#)
- Shortcuts [1076](#)
- SMS [1090](#)
- System Configuration [1074](#)
- type libraries [1387](#)
 - checking [1387](#)
 - checking with QualityMonitor [1387](#)
- Type Libraries view [1422](#)
- type library file path [1189](#)
 - specifying in Visual Studio C++ [1189](#)

U

- U [426](#)
- UI sequences [1044](#)
 - hiding dialogs during [1044](#)
- universal transforms [1015](#), [1064](#)
 - creating [1015](#), [1064](#)
- Unpublished Packages dialog box [1323](#)
- update [295](#)
 - manually updating a subscription [295](#)
- Updated.isr [635](#)
- upgrading
 - 5.0 or 5.5 or 6.0 Application Catalogs [221](#)
 - pre AdminStudio 5.0 Application Catalogs [221](#), [386](#)
 - upgrading AdminStudio databases [221](#)
- Use Application Expiration [923](#)
- UseHKCUProxy [628](#)
- UseMergeModules [628](#)
- user accounts [179](#)
 - disabling a user account [161](#)
 - importing from Active Directory [179](#)
- user interface reference [1058](#)
- user roles [194](#)
- user-defined ACE rules [1182](#), [1184](#)
- user-defined ACEs [1183](#), [1191](#), [1192](#)
 - deleting [1192](#)
 - editing [1191](#)
- User-Defined Tests view [1431](#)
 - Test Case view [1431](#)
- users
 - adding to a publication access list [290](#)
 - deleting [162](#)
 - People Administration Page [176](#)
 - Person Details View [178](#)
- UseSrcFolder [628](#)

V

- validate options [1244](#)
 - ConflictSolver [1244](#)
- Validate Package [1112](#)
 - deleting ICE errors [1115](#)
 - suppressing ICE errors [1114](#)
 - viewing ICE error information [1113](#)
- Validate Tab [316](#), [1208](#)
- validation [93](#), [1016](#), [1109](#), [1110](#), [1111](#), [1112](#), [1116](#), [1117](#), [1215](#)
 - before import [1110](#)
 - changing default ConflictSolver options [1116](#)
 - changing the default file in ConflictSolver [1116](#)
 - ConflictSolver [93](#), [1109](#), [1110](#)
 - deleting ICE errors [1115](#)
 - during ConflictSolver import [1111](#)
 - Explanation dialog [1235](#)
 - handling invalid packages in ConflictSolver [1117](#)
 - in ConflictSolver [1110](#)
 - suppressing ICE errors [1114](#)
 - viewing ICE error information [1113](#)
- Validation tab [1057](#)
 - in ConflictSolver Options dialog [1244](#)
 - launching Direct Editor [1057](#)
- Validation view [1215](#)
- Validation Wizard [1110](#), [1258](#), [1259](#)
 - MSI Source Information panel [1259](#)
 - Summary panel [1259](#)

Index

- using [1110](#)
- variables [145](#)
 - DevLocation [145](#)
 - InstallLocation [145](#)
 - ProjectName [145](#)
 - SharedPoint [145](#)
 - SourcePackage [145](#)
 - TargetDir [145](#)
 - TargetFileName [145](#)
- verifying component deployment [1397](#)
- verifying files [1399](#)
 - using MSI Doctor in QualityMonitor [1399](#)
- version [516](#)
- version management [250](#)
 - in Software Repository [256](#)
 - using the Software Repository [250](#)
- View Bar [1062](#)
- View Settings tab [1064](#)
- viewing tool properties [149](#)
- Views [596](#)
- views [1067](#), [1068](#), [1071](#), [1072](#), [1073](#), [1075](#), [1076](#), [1081](#), [1082](#), [1083](#), [1085](#), [1086](#), [1087](#), [1088](#), [1089](#), [1090](#), [1211](#), [1213](#), [1215](#), [1217](#), [1218](#), [1219](#), [1220](#), [1222](#)
 - Additional Tools [1090](#)
 - Associated Patches [1219](#)
 - Conflicts [1213](#), [1222](#)
 - ConflictSolver [1211](#)
 - Dependencies [1218](#)
 - Dialogs [1086](#)
 - Direct Editor [1090](#)
 - Features [1073](#)
 - Files and Folders [1075](#)
 - Help [1071](#)
 - INI Files [1081](#)
 - Location [1090](#)
 - Merge Module [1219](#)
 - NT Services [1083](#)
 - ODBC Resources [1082](#)
 - OS Snapshot [1219](#)
 - Other Setup Types [1220](#)
 - Package [1089](#)
 - Package Preparation [1087](#)
 - Package Validation [1071](#)
 - Patch Impact [1217](#)
 - Postvalidation [1088](#)
 - Prevalidation [1071](#)
 - Product [1211](#)
 - Product Properties [1072](#)
 - Registry [1075](#)
 - Server Locations [1085](#)
 - Setup Properties [1086](#)
 - Setup.exe [1090](#)
 - Shortcuts [1076](#)
 - SMS [1090](#)
 - Tables [1219](#), [1220](#)
 - Validation [1215](#)
- virtual package
 - deleting Windows Installer package association [244](#)
- virtual package conversion
 - types of [648](#)
- virtual packages
 - deleting an association [244](#)
 - importing during import of Windows Installer package [239](#)
 - importing into Application Catalog [238](#), [243](#)
 - importing into Application Catalog after import of Windows Installer package [241](#)
 - manually associating with Windows Installer package [243](#)
 - testing prior to conversion [673](#)
- Virtual Suite conversion type [649](#)
- Virtual Suite option [700](#)
- virtualization [516](#), [532](#), [695](#)
 - about [766](#)
 - benefits of [647](#), [648](#), [767](#)
 - benefits of Citrix XenApp [666](#)
 - building a Citrix profile using Repackager [662](#), [668](#)
 - building a ThinApp application using Repackager [659](#)
 - capturing context [694](#)
 - command line [650](#)
 - context file [694](#)
 - converting Windows Installer package to Citrix profile [668](#)
 - converting Windows Installer package to ThinApp
 - application [668](#)
 - creating virtual applications [765](#)
 - customizing applications [765](#)
 - diagram of [766](#)
 - example diagram [647](#)
 - example of [767](#)
 - getting started [644](#)
 - in InstallShield [768](#)
 - including additional MSIs [836](#), [882](#), [908](#)
 - methods to convert Windows Installer packages [644](#)
 - overview of [646](#)
 - overview of Citrix profiles [660](#), [664](#), [939](#)
 - overview of Citrix XenApp [663](#), [937](#)
 - using InstallShield Editor virtual Assistants [650](#)
 - via command line [516](#)
 - Windows Installer Converter [668](#)
- virtualization Assistants
 - integration with Project Assistant and Installation Designer [768](#)

- navigating in [770](#)
 - opening the Installation Designer [770](#)
 - showing and hiding [771](#)
- visibility [1023](#)
 - changing for features [1021](#)
 - feature [1023](#)
- visible property [1023](#)
- visiblity [1021](#)
- VMware [621](#)
 - ThinApp application [610](#)
- VMware Repackaging Wizard [621](#)
 - VMware Virtual Machines panel [621](#)
 - Welcome panel [621](#)
- VMware ThinApp [610](#)
- VMware ThinApp application [610](#)
- VMware Virtual Machines panel [621](#)
 - VMware Repackaging Wizard [621](#)

W

- Warning Frequency [924](#)
- Warning Message [924](#)
- Warning Period [924](#)
- watermark [75](#)
- Web Service URL panel [1571](#)
- Welcome Panel [1569](#)
- Welcome panel [152](#), [153](#), [154](#), [500](#), [615](#), [619](#), [621](#), [1092](#), [1093](#), [1249](#), [1253](#), [1363](#), [1463](#)
 - Add Tool Wizard [152](#)
 - Application Isolation Wizard [1363](#)
 - Conflict Wizard [1249](#)
 - Distribution Wizard [1463](#)
 - Import INI File Wizard [1092](#)
 - Import REG File Wizard [1093](#)
 - Import Wizard [365](#)
 - in New Workflow Project Wizard [153](#)
 - Merge Wizard [377](#)
 - OS Snapshot Wizard [378](#)
 - Package Auto Import Wizard [374](#)
 - Patch Impact Analysis Wizard [1344](#)
 - Predeployment Analysis Preparation Wizard [1569](#)
 - Repackaging Wizard [500](#)
 - Rules Wizard [1253](#)
 - Setup Intent Wizard [619](#)
 - SmartScan Wizard [615](#)
 - VMware Repackaging Wizard [621](#)
 - Windows Installer Converter [696](#)
- Welcome to InstallShield Tuner [1068](#)
- Welcome to Tuner view
 - Create a new transform file option [1068](#)
- wfsp_ [388](#)
- wftbl [387](#)
- Where Clause Panel [1257](#)
 - Rules Wizard [1257](#)
- Where Clause tab [1230](#)
- Windows 2000 [1086](#)
 - Add/Remove Programs [1086](#)
- Windows Installer Converter [645](#), [668](#), [695](#)
 - Batch Conversion conversion type [650](#)
 - Batch Convert conversion mode [700](#)
 - benefits of using to create App-V applications [651](#)
 - bulk conversion of App-V applications [654](#)
 - command line parameters [693](#)
 - Conversion Mode panel [698](#)
 - conversion modes [648](#)
 - converting a directory of applications to a virtual suite [650](#)
 - converting a group of applications to a virtual suite [649](#)
 - converting to App-V applications [645](#)
 - converting to Citrix profiles [645](#)
 - creating 64-bit App-V applications [655](#)
 - custom rules [690](#)
 - custom rules reference [681](#)
 - Edit Windows Installer Package panel [705](#)
 - Evaluate packages for virtualization suitability option [700](#)
 - evaluating packages for virtualization suitability [650](#)
 - jumping to Citrix Assistant [705](#)
 - jumping to InstallShield Citrix Assistant [668](#)
 - location of generated files [664](#)
 - location of output [660](#)
 - msi2vp.exe [693](#)
 - overview of Citrix profiles [664](#)
 - Package Conversion panel [706](#)
 - Pre-Conversion Evaluation feature [673](#)
 - pre-conversion evaluation rules [690](#)
 - running from command line [668](#)
 - Select a Directory panel [701](#)
 - Select a Suite of Packages panel [702](#)
 - Select Conversion Target panel [697](#)
 - Select Windows Installer Package panel [700](#)
 - Single Application conversion mode [649](#), [699](#)
 - support for Windows services [658](#)
 - Test Packages panel [704](#)
 - testing [673](#)
 - testing App-V application using App-V Application Launcher [654](#), [691](#)
 - types of virtual package conversion [648](#)
 - using to convert a Windows Installer package to a virtual application [643](#)
 - using to create App-V application [651](#)
 - View Results button [705](#)
 - viewing test results [705](#)
 - Virtual Suite conversion mode [700](#)

Index

- Virtual Suite conversion type [649](#)
- vs. App-V Sequencer [651](#)
- Welcome panel [696](#)
- Windows Installer File Selection panel [1364](#), [1570](#)
 - Application Isolation Wizard [1364](#)
- Windows Installer isolated components [1356](#)
 - isolation method [1356](#)
- Windows Installer package [1017](#), [1018](#)
 - and .context.msi [694](#)
 - building in Repackager [532](#)
 - converting to Citrix profile using Repackager [668](#)
 - converting to ThinApp application using Repackager [668](#)
 - converting to ThinApp applications [645](#)
 - handing invalid [1018](#)
 - prevalidating [1017](#)
- Windows installer package
 - repackaging using the Repackaging Wizard [494](#)
- Windows Installer packages [232](#), [435](#), [436](#), [1117](#), [1131](#)
 - evaluating for virtualization suitability [650](#)
 - handing invalid in ConflictSolver [1117](#)
 - importing [230](#), [232](#), [234](#)
 - importing multiple using a configuration file [435](#)
 - importing simultaneously with merge modules [436](#)
 - resolving conflicts directly within [1131](#)
 - validating in ConflictSolver [1110](#)
- Windows registry
 - and App-V Assistant [807](#)
 - and Citrix Assistant [967](#)
 - and ThinApp Assistant [878](#)
- Windows Services
 - integration into an App-V application [658](#)
- Windows services
 - App-V support of [658](#), [782](#)
 - running within the virtual environment [658](#), [782](#)
- Windows Terminal Server
 - WTS01 ACE [1179](#)
 - WTS02 ACE [1179](#)
 - WTS03 ACE [1180](#)
 - WTS04 ACE [1181](#)
 - WTS05 ACE [1182](#)
- Windows XP [1086](#)
 - Add/Remove Programs [1086](#)
- WinINSTALL Conversion dialog box [595](#)
- WinINSTALL conversion problems [639](#)
 - Repackager [639](#)
- WinINSTALL project [531](#)
 - converting to Repackager project [531](#)
- Wise Installation project [531](#)
 - converting to Repackager project [531](#)
- wizards [152](#), [153](#), [1569](#)
 - AdminStudio Interface [152](#)
 - Application Isolation Wizard [1365](#)
 - Finishing INI File Import panel in Import REG File [1093](#)
 - Finishing Registry Import panel in Import REG File [1094](#)
 - Import Conflict Options panel in Import INI File [1092](#)
 - Import Conflict Options panel in Import REG File [1093](#)
 - Import INI File [1092](#)
 - Import INI File panel in Import INI File [1092](#)
 - Import REG File [1093](#)
 - Import Registry File panel in Import REG File [1093](#)
 - Location panel in Packaging [1094](#)
 - New Workflow Project Wizard [153](#)
 - OS Security Patch wizard [1325](#)
 - Package Summary panel in Packaging [1095](#)
 - Packaging [1094](#)
 - Patch Impact Analysis wizard [1344](#)
 - predeployment analysis [1569](#)
 - Predeployment Preparation Analysis [1569](#)
 - Setup.exe panel in Packaging [1095](#)
 - SMS panel in Packaging [1095](#)
 - Welcome panel in Import INI File [1092](#)
 - Welcome panel in Import REG File [1093](#)
- workflow [129](#)
 - example using the New Workflow Project wizard [129](#)
- Workflow Manager [139](#), [227](#)
 - and extended attributes [227](#)
 - integration with AdminStudio projects [139](#)
 - logging in as a guest [174](#)
 - tables [402](#)
- Workflow Selection panel [154](#)
 - in New Workflow Project Wizard [154](#)
- Workflow tables [401](#)
- workflow tasks [120](#), [138](#)
 - associating tools with [120](#)
- Workflow Templates tab [125](#), [126](#)
 - adding tools from [126](#)
 - reordering tasks in [125](#)
- workflows [122](#), [123](#), [124](#), [127](#), [129](#), [131](#), [138](#), [154](#)
 - creating [122](#)
 - creating new [123](#), [127](#)
 - deleting [124](#)
 - editing [122](#)
 - filtering [123](#)
 - New Workflow Project Wizard [154](#)
 - renaming [123](#)
 - saving [129](#)
 - tasks [138](#)
- Workflows Templates tab [138](#)
- Write Copy isolation option [870](#)
- WTS01 ACE [1179](#)
- WTS02 ACE [1179](#)
- WTS03 ACE [1180](#)

WTS04 ACE [1181](#)

WTS05 ACE [1182](#)

X

XenApp. See Citrix XenApp

XML file [225](#)

as extended attributes description [225](#)

Z

ZENworks [1452](#), [1456](#)

using Distribution Wizard to prepare packages for

desktop application distribution [1452](#)

using Distribution Wizard to prepare packages for server

distribution [1456](#)

ZENworks Configuration Management

preparing for distribution [1447](#)

ZENworks Configuration Management Server Login [1487](#)

ZENworks Desktop Application Panel

Distribution Wizard [1480](#)

ZENworks Desktop Application panel [1480](#)

ZENworks Desktop Management Agent

installing [528](#)

ZENworks projects

.aot and .axt files [527](#)

application object template files [528](#)

converting using Repackager [527](#)

ZENworks Server Distribution - Distributor Panel [1483](#)

Distribution Wizard [1483](#)

ZENworks Server Distribution - Object Panel [1482](#)

