

HP Client Automation

Administrator

for the Windows[®], Linux, and Macintosh operating systems

Software Version: 7.50

User Guide

Manufacturing Part Number: None

Document Release Date: May 2009

Software Release Date: May 2009



i n v e n t

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

© Copyright 2007-2009 Hewlett-Packard Development Company, L.P.

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

Microsoft®, Windows®, and Windows® XP are U.S. registered trademarks of Microsoft Corporation.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

PREBOOT EXECUTION ENVIRONMENT (PXE) SERVER
Copyright © 1996-1999 Intel Corporation.

TFTP SERVER

Copyright © 1983, 1993

The Regents of the University of California.

OpenLDAP

Copyright 1999-2001 The OpenLDAP Foundation, Redwood City, California, USA.
Portions Copyright © 1992-1996 Regents of the University of Michigan.

OpenSSL License

Copyright © 1998-2001 The OpenSSLProject.

Original SSLeay License

Copyright © 1995-1998 Eric Young (eay@cryptsoft.com)

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Documentation Updates

The title page of this document contains the following identifying information:

Software Version number, which indicates the software version.

- The number before the period identifies the major release number.
- The first number after the period identifies the minor release number.
- The second number after the period represents the minor-minor release number.

Document Release Date, which changes each time the document is updated.

Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition, visit the following URL:

<http://h20230.www2.hp.com/selfsolve/manuals>

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

<http://h20229.www2.hp.com/passport-registration.html>

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Table 1 indicates changes made to this document.

Table 1 Document Changes

Chapter	Version	Changes
Chapter 3	7.50	Page 54, To allow deployment of applications that require user interaction on Vista devices use the "runasuser" modifier for the Method to Install Resource method.
Chapter 3	7.20	Page 73, Publishing in Component Select Mode : added information about publishing for thin clients.
Chapter 3	7.20	Page 73, Publishing in Component Select Mode : Updated this section to include more information about verify and delivery options when publishing registry keys and links.
Chapter 3	7.50	Page 79, To allow deployment of applications that require user interaction on Vista devices use the "runasuser" modifier for the Method to Install Resource method.

Chapter	Version	Changes
Chapter 3	7.50	Page 88, Publishing Windows Installer Files : The Administrator Publisher can identify VMware ThinApp Isolated Applications when it is publishing a Windows Installer application and handles it accordingly.
Chapter 3	5.10 Jan 31, 08	Page 100, Command Line Publishing : changed the <code>test.cfg</code> command line to <code>nvdtk.exe pubport.tkd -cfg "c:\program files\test.cfg" -user USER -password PASSWORD</code> .
Chapter 3	5.10 Jan 31, 08	Page 100, in Table 17 : added more setting information for the parameter <code>msi {body}</code> . Corrected the setting names for <code>additionalfiles</code> and <code>connectaiptosvc</code> .
Chapter 3	5.10 Jan 31, 08	Page 102: added a closing bracket to the end of the sample configuration file.
Chapter 3	5.10 Jan 31, 08	Page 103, Defining which File to Publish on the Command Line : changed the sample command line to use with the <code>-file</code> parameter to <code>nvdtk.exe pubport.tkd -cfg "c:\program files\test.cfg" -file "c:\program files\testfile" -user USER -password PASSWORD</code>
Chapter 3	5.10 Jan 31, 08	Page 103, in Publishing Multiple .msi Files (Batch Mode) : added more details about how directories are scanned and more details about the files contained in designated location
Chapter 3	5.10 Jan 31, 08	Page 104, in Publishing Multiple .msi Files (Batch Mode) : changed the sample command line that could be used to run the Publisher that designates the location to scan to <code>nvdtk.exe pubport.tkd -batchmsi "c:\msifileslocation" -user USER -password PASSWORD</code> .
Chapter 3	5.10 Jan 31, 08	Page 104, in Publishing Multiple .msi Files (Batch Mode) : changed the name of the Publisher log file from <code>Publisher.log</code> to <code>pubport.log</code> .

Chapter	Version	Changes
Chapter 3	5.10 Jan 31, 08	Page 106, in Customizing Configuration File Values with Symbolic Substitution : the Sample imbedded MSI Properties Configuration File has been corrected. Do not use previously published version of this file, as they do not work.
Chapter 3	5.10 Jan 31, 08	Page 219:, redefined several menu options on the Feature Set Editor including Maximum Size, Total Size, and Selected Size.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online software support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support site to:

Search for knowledge documents of interest

Submit and track support cases and enhancement requests

Download software patches

Manage support contracts

Look up HP support contacts

Review information about available services

Enter into discussions with other software customers

Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in.

Many also require an active support contract. To find more information about support access levels, go to the following URL:

http://h20230.www2.hp.com/new_access_levels.jsp

To register for an HP Passport ID, go to the following URL:

<http://h20229.www2.hp.com/passport-registration.html>

Contents

1	Introduction	15
	Introduction	15
	Using this Guide with Core and Satellite Servers	15
	Audience	16
	Prerequisites	16
	About the Administrator	16
	Chapter Summaries	17
	Related Documents	17
2	Packaging and Publishing	19
	Introduction	20
	Packaging versus Publishing	20
	Packaging	20
	Packaging Considerations Checklist	22
	What Product Do I Use?	25
	Windows	26
	Linux	26
	Macintosh	26
	Batch Publisher	26
	Native Packaging	27
3	Administrator for Windows	29
	Installation	30
	System Requirements	30
	Platform Support	30
	About the Installation Files	30
	setup.exe	30
	HPCAЕ-ADMIN72.MSI	31

Installing the Administrator for Windows	31
Installation Wizard.....	31
Installing with a Command Line.....	34
Changing the Administrator Installation.....	36
Using the Installation Wizard to Change the Installation	37
Using a Command Line to Change the Installation	37
Agent Explorer	39
Starting the Agent Explorer.....	39
Menus in the Agent Explorer Window.....	40
File Menu	41
Edit Menu	41
View Menu	42
Window Menu	42
Help Menu.....	43
Object View Window	43
Object Menu	43
Variable Menu	44
Instance Menu	45
Options Menu.....	45
Redirecting the CSDB Editor to another Database	46
Packager	47
Set Package Properties	47
Agent Management Tab.....	48
Data Options Tab.....	51
Agent Behaviors Tab	53
Component Select Mode.....	55
Installation Monitor Mode.....	55
Recommendations.....	56
Using Installation Monitor Mode to Create a Package.....	57
Publisher	72
Overview	72
Batch Mode	72
Standard Publishing Steps	72
Publishing in Component Select Mode	73
Publishing Windows Installer Files.....	88
VMware ThinApp Isolated Applications.....	88
Upgrading an Existing Package using an .MSP File	90
Advanced Publishing Mode.....	91
Basic Publishing Mode	98
Command Line Publishing	100

Defining which File to Publish on the Command Line	103
Publishing Multiple .msi Files (Batch Mode).....	103
Using Multiple Configuration Files.....	104
Including Transform (.mst) Files.....	105
Customizing Configuration File Values with Symbolic Substitution	106
Embedded MSI Properties and Extensions for Windows Installer User-Defined Properties	108
Publishing Hardware Configuration Elements.....	108
Screen Painter	109
File Menu.....	109
Edit Menu	110
Font Menu	110
Controls.....	111
Controls Toolbox	111
Properties Box.....	112
Controls Menu	113
Alignment Menu	121
System Menu	123
Help Menu.....	124
CSDB Editor	125
Getting Started.....	125
The CSDB Editor Window	126
Setting Preferences: The Options Dialog Box.....	129
Navigating the CSDB Editor	137
Inheritance in the CSDB.....	140
Managing Domains, Classes, and Instances	141
Working with Domains	141
Working with Classes.....	142
Working with Instances	155
Password Authentication.....	162
EDMSIGN and EDMSIGNR.....	162
Without Password Authentication Enabled	163
With Password Authentication Enabled.....	164
Enabling Password Authentication.....	167
Changing Passwords	169
Access Rules and Administrators	169
Creating Services with the New Application Wizard	170
Defining MSI Basic Behaviors in the New Application Wizard	175
The Version Group Editor	177
Service Optimization.....	177

Using the Optimize Service Option	178
Service Groups	187
Component Enhancements	188
Prerequisites	188
Creating a Master Service for a Service Group	189
Notify	192
Filtering—Viewing Portions of the CSDB	194
Permanent Filters	195
Temporary Filters	197
Viewing the Base or Null Instance	198
Appending Filter Results	198
Filtering Component Class Instances	198
Express Filters	199
Maintaining Connection Rules in the CSDB	202
Connection Rules Overview	202
Enabling Connections in New Classes	206
Managing WI-enabled Applications	211
Feature Set Editor Overview	211
Accessing the Feature Set Editor	212
About the MSI Feature Set Editor	212
Windows Add or Remove Programs Settings	219
Copying a Windows Installer Enabled Package	221
Managing Feature Sets in Native Mode	222
The Registry File Editor	223
Accessing the Registry File Editor	223
Set Properties Shortcut Menu	225
Editing a Registry Key	227
AMP Editor	230

4 Administrator for Linux and Macintosh 231

Introduction	232
Installation	232
System Requirements	232
Platform Support	232
Prerequisites	232
Troubleshooting	233
Recommendations	233
Installation Methods	233
Installing the Administrator for LINUX or Macintosh	234

Graphical Installation	234
Command-Line Installation.....	236
Agent Explorer	237
Opening the Agent Explorer.....	237
Menus in the Object List Window	238
Object Menu	238
Edit Menu	240
View Menu	240
Table Window Menu.....	240
Menus in the Object View Window	240
Object Menu	240
Variable Menu	241
Heap Menu.....	243
Packager	244
Publisher	244
Prerequisites	245
Publishing.....	245
UNIX File Resources (UNIXFILE).....	247
Published Owner, Group, and Permission Considerations	248
A Modifying Component Instances (Non-Production Only)	251
When to Use these Commands	252
Accessing the Add, Edit, and Replace Component Commands.....	253
Editing a Component	254
Using the Editor Selector Dialog Box.....	257
Replacing Component Data	258
Adding a Component.....	260
B Naming Conventions	263
Categorizing Information.....	263
Naming Conventions for the POLICY Domain	264
Naming Conventions for the SOFTWARE Domain	264
Index	267

1 Introduction

Introduction

The *HP Client Automation Administrator User Guide (Admin Guide)* describes the six features of the HP Client Automation Administrator (Administrator) for the platforms for which they are available including Windows, Linux, and Macintosh:

- Agent Explorer (Windows, Linux)
- AMP Editor (Windows)
- CSDB Editor (Windows)
- Packager (Windows)
- Publisher (Windows, Linux, and Macintosh)
- Screen Painter (Windows)

The details of the AMP Editor are not discussed in this book. Refer to the *HP Client Automation Application Management Profiles User Guide (AMPs Guide)* for this information.

Using this Guide with Core and Satellite Servers



If your environment uses Core and Satellite servers, first read the *Core and Satellite Servers Getting Started Guide* as the installation, configuration, and troubleshooting information in that guide may override the information in this guide.

Audience

Administrators of HP Client Automation (HPCA) products can use this book to learn how to install the Administrator, and how to use five of its six features.

Prerequisites

Before you use the Administrator you should be familiar with the HP Client Automation Configuration Server (Configuration Server) and the HP Configuration Management Configuration Server Database (CSDB). Refer to the *HP Client Automation Configuration Server User Guide (Configuration Server Guide)* and the *HP Client Automation Configuration Server Database Reference Guide (CSDB Reference Guide)*.

About the Administrator

The Administrator contains a set of tools that you use to manipulate the contents of the Configuration Server Database (CSDB).

- **Agent Explorer**
allows you to create and edit Client Automation objects.
- **AMP Editor**
allows you to rapidly deploy and manage software products that are typically required on desktop clients, such as Microsoft Office 2007, Symantec Antivirus, and Citrix Presentation Agent among others. Refer to the *AMPs Guide*.
- **CSDB Editor**
allows you to inspect and manipulate the contents of the CSDB.
- **Packager**
allows you to use a packaging method called Installation Monitor Mode when you are not familiar with all the components that should be part of a package. This method performs a scan before and after you install the software. The difference between the two scans becomes the contents of the package.

- **Publisher**
allows you to use Component Select Mode to create a package and to then add that package to the CSDB. When you use this method, you select each component that you want to add to the package.
- **Screen Painter**
allows you to design and create dialog boxes.

Chapter Summaries

- Chapter 2, [Packaging and Publishing](#)
describes what the packaging and publishing processes are, and which product to use for your specific needs.
- Chapter 3, [Administrator for Windows](#)
describes how to install the Administrator for Windows and how to use its features.
- Chapter 4, [Administrator for Linux and Macintosh](#)
describes how to install the Administrator for Linux and Macintosh and how to use its features.
- Appendix A, [Modifying Component Instances](#)
describes how to dynamically edit the distributed contents of a file from within the Admin CSDB Editor.
- Appendix B, [Naming Conventions](#)
describes how naming conventions can help you organize the data stored in the CSDB.

Related Documents

HP Client Automation Application Manager and Application Self-service Manager Installation and Configuration Guide (Application Manager and Application Self-service Manager Guide)

HP Configuration Management Batch Publisher Installation and Configuration Guide (Batch Publisher Guide)

HP Client Automation Configuration Server Database Reference Guide (CSDB Reference Guide).

HP Client Automation Configuration Server User Guide (Configuration Server Guide)

HP Client Automation Essentials Guide (Essentials Guide)

HP Configuration Management Extensions for Windows Installer Getting Started Guide (Extensions for WI Getting Started Guide)

HP Configuration Management Extension for Windows Installer User Guide (Extensions for WI Guide)

HP Client Automation Knowledge Base Server Installation and Configuration Guide (KB Server Guide)

HP Client Automation OS Manager Hardware Configuration User Guide (Hardware Configuration Guide)

HP Client Automation OS Manager System Administrator Guide (OS Manager Guide)

HP Client Automation Application Management Profiles User Guide (AMPs Guide)

2 Packaging and Publishing

At the end of this chapter you will:

- Understand the differences between packaging and publishing data.
- Understand which Client Automation products to use to package data.
- Understand which Client Automation products to use to publish data.

Introduction

In Client Automation, you must process data in two ways before you can deploy them in your environment: you must first package them, then publish them to the Configuration Server Database (CSDB). Several Client Automation products perform these operations. The one you use depends on the type of data you want to deploy and on which operating system the HP Client Automation Administrator (Administrator) is running.

This chapter explains what the packaging and publishing processes are, and which product to use for your specific needs.

Packaging versus Publishing

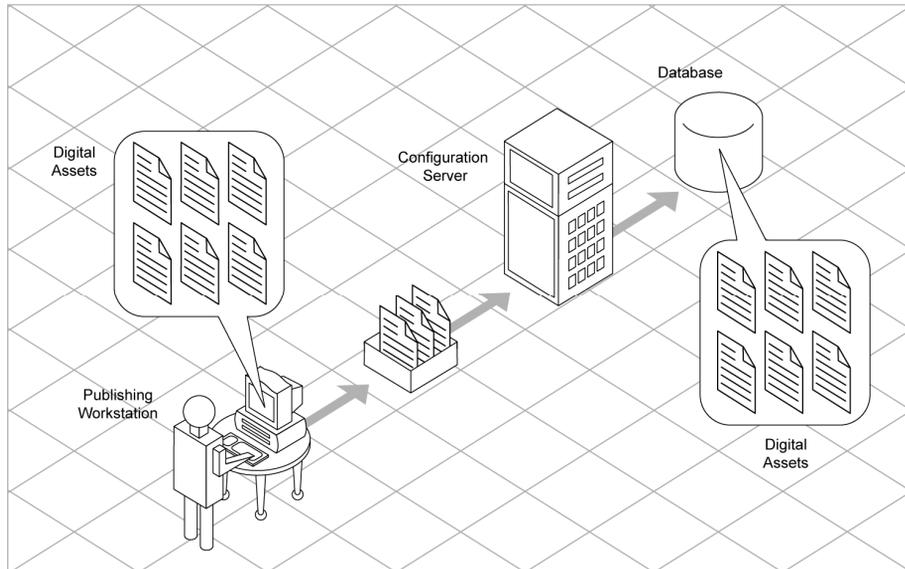
In Client Automation, packaging and publishing data are two distinct processes. The data that you distribute can vary greatly—from a single data file, such as a company telephone list, to an entire application suite, such as Microsoft Office.

- **Packaging**
is the process of identifying the components of the software that you want to manage, and organizing them into **packages**. Packages contain the files, shortcuts, links, and/or registry entries that make up the software.
- **Publishing**
is the process of importing a package and its imbedded information into the CSDB. A package must be published before its content can be distributed and deployed into your environment.

Packaging

The two primary ways of creating packages are Installation Monitor Mode and Component Select Mode. Several factors influence which mode you will use: the complexity of the application, your knowledge of the application's structure, and whether you would like to use the application's native installation capabilities.

Figure 1 Packaging data



- **Installation Monitor Mode is a feature of the Packager.**
In Installation Monitor Mode, the Packager determines what to package by scanning the computer before and after installing the software. It **differences** the before and after scans to determine what changes were made to the computer. These differences become the package that you promote to the CSDB. We recommend this mode for packaging when you do not know all of the components that make up the application. This process is described in detail in [Packager](#) on page 47.
- **Component Select Mode is a feature of the Publisher.**
In Component Select mode, you select the individual components that make up the application, such as files, directories, registry entries, and links. We recommend this mode for packaging simple data or applications where you can easily identify all the components in the package. This process is described in detail in [Publisher](#) on page 72.

After you create a package, you **publish** it to the CSDB. The package is copied to the CSDB and several instances are created, as described below.

- An Application Packages (PACKAGE) instance that represents the promoted package.
- One File Resources (FILE) instance for each file in the package.

- One Desktop (DESKTOP) instance for each program group, link, and shortcut in the package.
- One Path (PATH) instance for each unique path to one or more components on the computer where the software is installed.
- One Registry Resources (REGISTRY) instance for each hive in the package.



Each instance described above is stored in one of the default classes in the SOFTWARE Domain, such as the PACKAGE or FILE Class. You can also add your own classes, such as a DLL class, to the CSDB. See [Adding a Class](#) on page 142 for information on how to add a class.

After promoting the package, use the CSDB Editor to create a service and assign policies. See the Entitlement chapter in the *HP Client Automation Application Manager and Application Self-service Manager Installation and Configuration Guide (Application Manager and Application Self-service Manager Guide)* for more information.

Packaging Considerations Checklist

Before packaging your data, there are many things that you should consider.

General

- What do you know about the data to be packaged?
 - If it is a simple package and you can easily identify all its components, use Component Select Mode in the Publisher.
 - If it is a more complex package and you do not know what files to include, use Installation Monitor Mode in the Packager.
- What is the name of the package going to be? Follow your naming conventions. See Appendix B, [Naming Conventions](#).
- Do you have a unique session ID? Follow your conventions.

System Configuration

See [Set the Required System Configuration](#) on page 59.

- What operating systems are your target computers (workstations or servers) using?
- How much RAM is needed to handle the data?

- What is the minimum processor needed to handle the data?

64-Bit Applications

Use Packager on a 64-bit computer to package 64-bit applications, and on a 32-bit computer to package 32-bit applications, regardless of the target computer's bit architecture. This avoids accessing both 32-bit and 64-bit registry and file systems during one packaging session.

32-bit applications being deployed to the `Program Files` directory will not be redirected to the `Program Files (x86)` directory by the Application Manager on a 64-bit machine. This is a convention implemented through environment variables, and not a hard requirement. There will be no adverse effect to installing 32-bit applications in the `Program Files` directory, unless you are deploying a 64-bit version of the application to the same device and same `Program Files` directory. If this situation arises, there are two methods to resolve it:

- Use the CSDB Editor to copy the 32-bit package and change the `Program Files PATH` instances to reference `Program Files (x86)`.
- Convert the application to an MSI install and allow MSI to determine the correct `Program Files` location.

The packaging and publishing tools on a 64-bit operating system automatically generate a ZSTOP expression. Two examples are:

- `WORDPOS(EDMGETV(ZMASTER,ZOSTYPE),'WINX64_NT')=0`
- `WORDPOS(EDMGETV(ZMASTER,ZOSTYPE),'WINIA64_NT')=0`

Activation Options

- When do you want to activate the application - immediately on distribution or later? See [Set Date and Time Constraints](#) on page 60.
- Which version of the application do you want to distribute, and when do you want to activate it? See the *Preparing Services* chapter in the *Application Manager and Application Self-service Manager Guide*, for more information.

Data Options

See [Data Options Tab](#) on page 51.

- What type of compression do you want to use?
- Will your data be stored in the CSDB or on a proxy server?

- Do you want to preserve the security settings for your files and folders that are being deployed to NTFS volumes?
- Are you distributing maintenance to the Application Self-service Manager agent?
- How do you want to promote the resources? Will you use the force lock method? See [Force lock method](#) on page 52.
- Are you sending out an update and, therefore, only want to deploy the changes?

Verify Options

See [Agent Management Tab](#) on page 48.

- Do you want to use the standard default verification options?
- Is this a first time installation? Is there anything that you need to verify?
- When deploying files, what types of statistics do you want to check – date, time, size?
- Do you want to update all files, or only newer files?
- If a file already exists, do you want to deploy it again to overwrite any changes that may have been made?
- Do you want to use the internal version to determine whether a file should be deployed?

Delivery Options

See [Agent Management Tab](#) on page 48.

- Do your files or methods need to be deployed in a particular order?
- Is the data mandatory or optional? You can only deliver mandatory files if the Application Manager is installed.
- Do you want the data deployed under the user or machine context?

Agent Behaviors

See [Agent Behaviors Tab](#) on page 53.

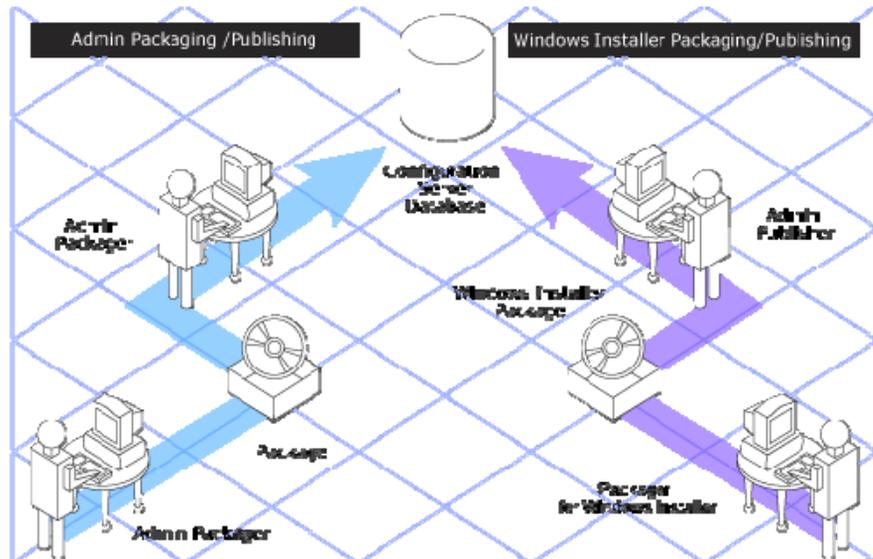
- After the file is deployed, do you want to run any methods? If so, what are they?
- Does anything need to happen to enable the file once it is deployed? If so, what method will you run to enable it?

- If the subscriber is no longer subscribed to the software, do you want to delete the file?
- Do you want to compare the old and new version of the file that you are deploying? If so, what method do you want to use?

What Product Do I Use?

HP Client Automation has several tools that you can use to package and publish data. Identify the different types of resource packages that you want to create and publish, then choose your Client Automation packaging and publishing products as needed.

Figure 2 Packaging/Publishing options



- If you want to package Windows Installer files, use the Packager for Windows Installer. (Refer to the *Extensions for Windows Installer User Guide*).
- If you want to publish Windows Installer files, OS images, or hardware configurations, use the Publisher (see [Publisher](#) on page 72).
- To package and publish all other types of files and applications for Windows, use the Packager.

Windows

- **Packager**
Use this tool to package and publish Windows applications using Installation Monitor Mode.
- **Publisher**
Use this tool to package and publish OS images, HW configurations, and MSI files, and to use Component Select Mode.
- **Batch Publisher**
Use this tool for automated publishing. See [Batch Publisher](#) below.
- **Packager for Windows Installer**
Use this tool, which is a component of the Extensions for Windows Install, to create MSI packages

Linux

There are two packaging/publishing tools available for Linux.

- **Publisher**
Use this tool to package and publish data. This tool contains only the Component Select mode of publishing.
- **Batch Publisher**
Use this tool for automated publishing as well as publishing of native Linux software packages.

Macintosh

There is only one Administrator tool, the Publisher, available for the Macintosh. Use it to package applications in Component Select Mode and publish them to the CSDB.

Batch Publisher

The Batch Publisher is a command-line alternative to using Component Select Mode. It offers an automated, repeatable command-line process to create packages and store them in the CSDB. Refer to the *HP Configuration Management Batch Publisher Installation and Configuration Guide (Batch Publisher Guide)*.

The Batch Publisher can:

- Search for files on multiple drives/file systems.
- Search for and publish files from any mapped file/drive system.
- Be configured to limit the subdirectories that are searched.
- Include or exclude at the file level.
- Select files by type.

The Batch Publisher can also accommodate frequent patching of internal applications. Its capacity to revise content is reliable, and can be designed to perform continuously, at designated times, and at predetermined intervals. The Batch Publisher can be easily executed from within any script or code capable of calling a command prompt.

Native Packaging

Native Packaging is a feature of the Batch Publisher specifically designed for UNIX environments. Native Packaging is a command-line driven content-publishing tool that supports native HP-UX and Solaris software. Native Packaging is installed during the regular installation of the Batch Publisher on a UNIX system.

Native Packaging explores UNIX native software depots, searches for available native packages, and publishes wrapped native packages to the Configuration Server. Native Packaging publishes all the information that you need for immediate installation of native software to end agents. When the Application Manager agent is installed, a Tcl script is included in the `IDMSYS` directory that is required when packages that are published using Native Packaging are deployed. For more information, refer to the *Batch Publisher Guide*.



The Batch Publisher is an optional feature available from HP. Contact your sales representative for more details.

3 Administrator for Windows

At the end of this chapter, you will:

- Understand how to install the HP Client Automation Administrator (Administrator) and its feature components.
- Understand how to use the Agent Explorer to create and edit objects.
- Understand how to use the Packager to package and publish data to the CSDB.
- Understand how to use the Publisher to package and publish data to the CSDB.
- Understand how to use the Screen Painter to design and create dialog boxes.
- Understand how to use the CSDB Editor to create and maintain a desired state.

Installation

The Administrator installation program uses Microsoft Windows Installer. The program consists of one MSI package with six feature sets— Agent Explorer, Packager, Publisher, Screen Painter, CSDB Editor, and AMP Editor.

System Requirements

- Clean computer. (A **clean computer** is a computer with only the target subscriber's operating system installed.)
- Windows 2000, Server 2003, XP, or Vista (x86 or x64 where applicable).
- TCP/IP connection to the HP Client Automation Configuration Server (Configuration Server).
- Minimum screen resolution of 800 x 600.
- For Windows 2000, Server 2003, XP, or Vista, you must have administrator rights to the computer to install the Administrator.

Platform Support

For detailed information about supported platforms, see the HPCA 7.20 Release Notes that accompany this release.

About the Installation Files

This section describes the files that you use for installation.

setup.exe

`setup.exe` is located on the Client Automation infrastructure media in the Configuration Server\management_infrastructure\administrator_workstation\win32\ folder. It accepts any command line parameters and passes them to Windows Installer.

You can also create a Windows Installer Administrative Installation Point (AIP) for network installations.



A Windows Installer Administrative Installation Point (AIP) is also known as an Administrative Control Point (ACP).

The AIP starts Windows Installer and passes command line parameters to it. To create the Windows Installer Administrative Installation Point (AIP) in a specified target directory, type:

```
SETUP.EXE /a TARGETDIR=drive:\targetdirectory /qb
```

The target directory contains `HPCAE-ADMIN72.MSI`, the installation folders, and `setup.exe`.

HPCAE-ADMIN72.MSI

`HPCAE-ADMIN72.MSI` is the MSI database file, which contains the default configuration information for the installation. This file is stored on the Client Automation infrastructure media in the Configuration Server `\management_infrastructure\administrator_workstation\win32` folder.

Installing the Administrator for Windows

This section describes how to install the Administrator for Windows.



The Configuration Management 4.2i or higher Administrator is not compatible with Configuration Management 4.0 or lower agent. Agent objects look like they are corrupt. Installing a 4.2i or 5.0 Administrator on a 4.0.x agent machine adds a `CENCODE` variable to the `LICENSE.EDM` object. This tells `NVDKIT` to create version 4 objects, which look corrupt to a version 4.0 agent. You cannot have a 4.2i or higher Administrator and a 4.0 or lower agent co-exist on the same machine. Both agent and Administrator need to be at the same level. To get the 4.0 agent working again, remove the `CENCODE` variable from the `LICENSE.EDM` object, and remove the Administrator from the machine.

Installation Wizard

You can use the Installation Wizard to install the Administrator for Windows.

To use the Installation Wizard to install the Administrator

- 1 From the folder on the installation media that contains the Administrator installation files, run `setup.exe`. The Administrator Installation Wizard opens.
- 2 Click **Next**. The License Agreement window opens.
- 3 Read and accept the license agreement, then click **Next**.

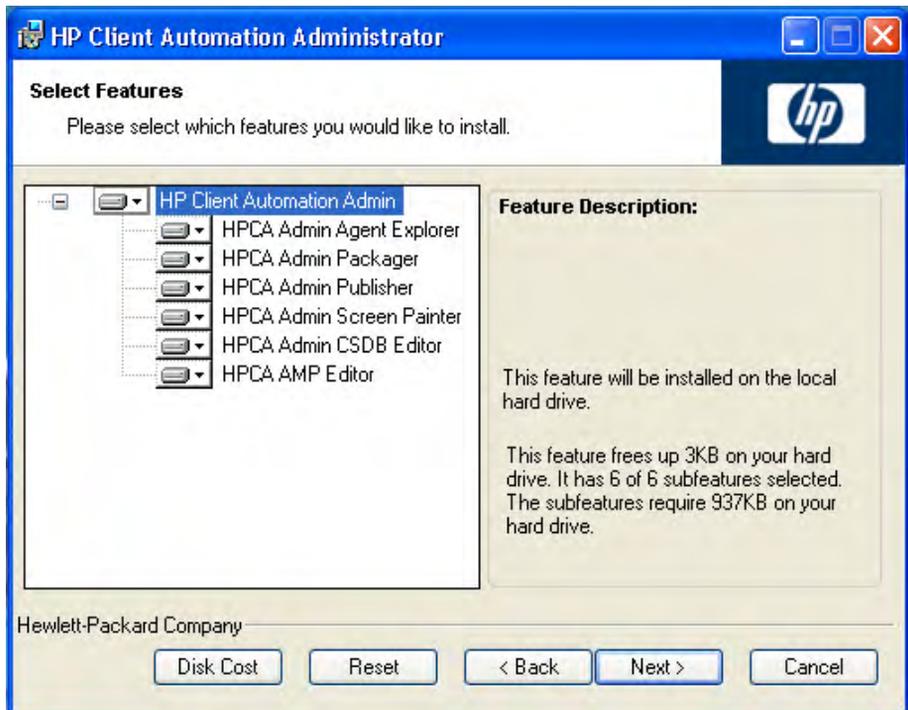
If a Client Automation agent is not installed on the computer, the Destination Folder window opens.

- ▶ If a Client Automation agent is already installed on the computer, the Destination Folder window will not open and the Administrator will be installed in the same location as the Client Automation agent.

If you want to select a different destination for the Administrator, click **Browse**, and then navigate to the appropriate destination folder, then click **OK** to continue.

- 4 Click **Next**. The Configuration Server window opens.
- 5 In the IP Address text box, type the IP address of the Configuration Server.
- 6 In the Port text box, type the port number (default is 3464).
- 7 Click **Next**. The License window opens.
- 8 Enter a valid license file (for Starter, Standard, or Enterprise edition), or navigate to and select it.
- 9 Click **Next**. The Select Features window opens.
 - If you entered a Starter or Standard license file in step 8, you will have 2 features from which to choose: HPCA Admin Agent Explorer, and HPCA Admin Publisher.
 - If you entered an Enterprise license file in step 8, you will have 6 features from which to choose: HPCA Admin Agent Explorer, HPCA Admin Packager, HPCA Admin Publisher, HPCA Admin Screen Painter, HPCA Admin CSDB Editor, and HPCA AMP Editor.

- ▶ If you entered an Enterprise license file in step 8, the next window opens.



10 Click  to select the features that you want to install.

 If you want to set the same options for all of the features, you can click  next to HP Client Automation Admin and select the appropriate option to apply the setting to all features.

Each time you click , a shortcut menu for that feature opens.

11 From the shortcut menu, select an installation option. [Table 1](#) below contains descriptions of these options.

Table 1 Feature settings for the Administrator

Option	Description
Will be installed on local hard drive	Installs the top-level feature on the local hard drive, but not any sub-features.

Option	Description
Entire feature will be installed on local hard drive	Installs the entire feature, including sub-features. Note: In this installation program, selecting this option or the "Will be installed on local hard drive" option for any of the features results in the same installation because these features do not contain sub-features.
Entire feature will be unavailable	The feature will not be installed. If previously installed, this feature will be removed.

- 12 You can click **Disk Cost** to see an overview of the disk space needed for the installation.
- 13 Click **Next**. The Ready to Install the Application window opens.
- 14 Click **Install** to begin the installation. When the installation is done, the Administrator has been successfully installed window opens.
- 15 Click **Finish** to exit the installation.

Installing with a Command Line

You can also use a command line to install the Administrator. From the directory containing the Administrator installation files type a command line with the following syntax:

SETUP.EXE FeatureStateArgument=Feature1, Feature2

where *FeatureStateArgument* is one of the arguments in [Table 2](#) below and *Feature1* and *Feature2* are arguments in [Table 3](#) on page 35.

Table 2 Administrator feature state arguments

Specify the following arguments	To set the feature state
ADDLOCAL	Type a comma-delimited list of features that you want to set to "Will be installed on local hard drive."
ADDSOURCE	Type a comma-delimited list of features that you want to set to "Will be installed to run from network."

Specify the following arguments	To set the feature state
ADVERTISE	Type a comma-delimited list of features that you want to set to "Feature will be installed when required."
REMOVE	Type a comma-delimited list of features that you want to set to "Entire feature will be unavailable." This only removes the features—not the entire product. Therefore, if you use the REMOVE property and type each of the feature names, the core product will still be stored on your computer. If you want to remove the entire product, type REMOVE=ALL.

Table 3 Admin features arguments

Admin Feature	Feature argument
Agent Explorer	NVDINSTALLCLIENTEXPLORER
Packager	NVDINSTALLPACKAGER
Publisher	NVDINSTALLPUBLISHER
Screen Painter	NVDINSTALLSCREENPAINTER
CSDB Editor	NVDINSTALLSYSTEMEXPLORER
AMP Editor	NVDINSTALLAMPEDITOR

Examples

If you want to install only the Packager on a computer, type the following command line:

```
SETUP.EXE ADDLOCAL=NVDINSTALLPACKAGER
```

If you want to install the CSDB Editor and the Agent Explorer on a computer, type the following command line:

```
SETUP.EXE ADDLOCAL=  
NVDINSTALLSYSTEMEXPLORER,NVDINSTALLCLIENTEXPLORER
```



If you run the installation from a command line, be sure to pass the IP address for the Configuration Server to the installation. For example:

```
SETUP.EXE NVDOBJZMASTER_ZIPADDR=10.10.10.1
```

Additional Command Line Arguments

Table 4 below describes some additional arguments that you can pass to the installation program on the command line.

Table 4 Command line arguments

Argument	Description
/qn	Performs a silent installation.
/qb	Displays the progress bar only during the installation.
/l*v drive:\install.log	Creates a detailed Windows Installer log. Note: Using this option may impact the performance of the installation.
/a TARGETDIR= drive: \targetdirectory	Creates a Windows Installer AIP in the specified target directory. The target directory contains HPCAADMIN72.MSI, the installation folders, and setup.exe. After you create the AIP, you can run setup.exe and pass the appropriate command line parameters. This starts the Windows Installer and passes the specified parameters to it.
INSTALLDIR=	Specify the installation directory to be used. Use quotes for paths that contain spaces.

Changing the Administrator Installation

The Windows Installer installation program allows you to modify, repair, and remove the Administrator.

Using the Installation Wizard to Change the Installation

This section describes how to use the Installation Wizard to modify, repair, and remove (uninstall) the Administrator.

- ▶ To remove specific features of the Administrator, use the Modify option on the Application Maintenance window.

To use the Installation Wizard to change the Administrator installation

- 1 From the folder on the installation media that contains the Administrator installation files, double-click **setup.exe**. The Application Maintenance window opens.
- 2 Select the **Modify**, **Repair**, or **Remove** option.
- 3 Click **Next**.
 - If you select **Modify**, the Select Features window opens. Select the features that you want to modify, then click **Install**. See [Installation Wizard](#) on page 31 for information on feature settings.
 - If you select **Repair**, the Ready to Repair the Application window opens. Click **Install**.
 - If you select **Remove**, click **Remove**.
- 4 Click **Finish**.

Using a Command Line to Change the Installation

This section describes how to use a command line to modify, repair, and remove (uninstall) the Administrator using a command line.

To use a command line to modify the Administrator

- From the folder on the installation media that contains the Administrator installation files, type the following command line:

```
SETUP.EXE FeatureStateArgument=Feature1, Feature2
```

where *FeatureStateArgument* is one of the arguments in [Table 2](#) on page 34 and *Feature1* and *Feature2* are arguments in [Table 3](#) on page 35.

Examples

If you want to silently remove the CSDB Editor and Agent Explorer, type:

```
SETUP.EXE REMOVE=NVDINSTALLSYSTEMEXPLORER,  
NVDINSTALLCLIENTEXPLORER /qn
```

To install the Packager to the local hard drive, and to make the CSDB Editor and Agent Explorer unavailable, type the following command line:

```
SETUP.EXE ADDLOCAL=NVDINSTALLPACKAGER  
REMOVE=NVDINSTALLSYSTEMEXPLORER,NVDINSTALLCLIENTEXPLORER
```

Agent Explorer

This section describes how to create and edit Client Automation objects using the Agent Explorer.

The Agent Explorer is installed automatically with the Administrator. You can use the Agent Explorer as a diagnostic utility to view local objects on the desktop and, if desired, to edit existing objects or create new ones.

If you have multiple Client Automation agents that rely on a single file server in your environment, you can use the Agent Explorer to edit objects stored on that file server.

The Agent Explorer can also be used to edit objects on other Client Automation agent desktops. For diagnostic purposes, you can view and edit managed objects configured for other Client Automation agents to which you are connected on a local area network (LAN).



Client Automation screen objects can be viewed with the Agent Explorer, however, the actual screens are not displayed. Client Automation screens must be created and edited using the Screen Painter. Be careful not to corrupt the variables of screen objects while viewing them with the Agent Explorer.

Starting the Agent Explorer

The following instructions explain how to open and close the Agent Explorer.

To open the Agent Explorer

- From the **HP Client Automation Administrator** folder, click **Client Automation Administrator Agent Explorer**.

or

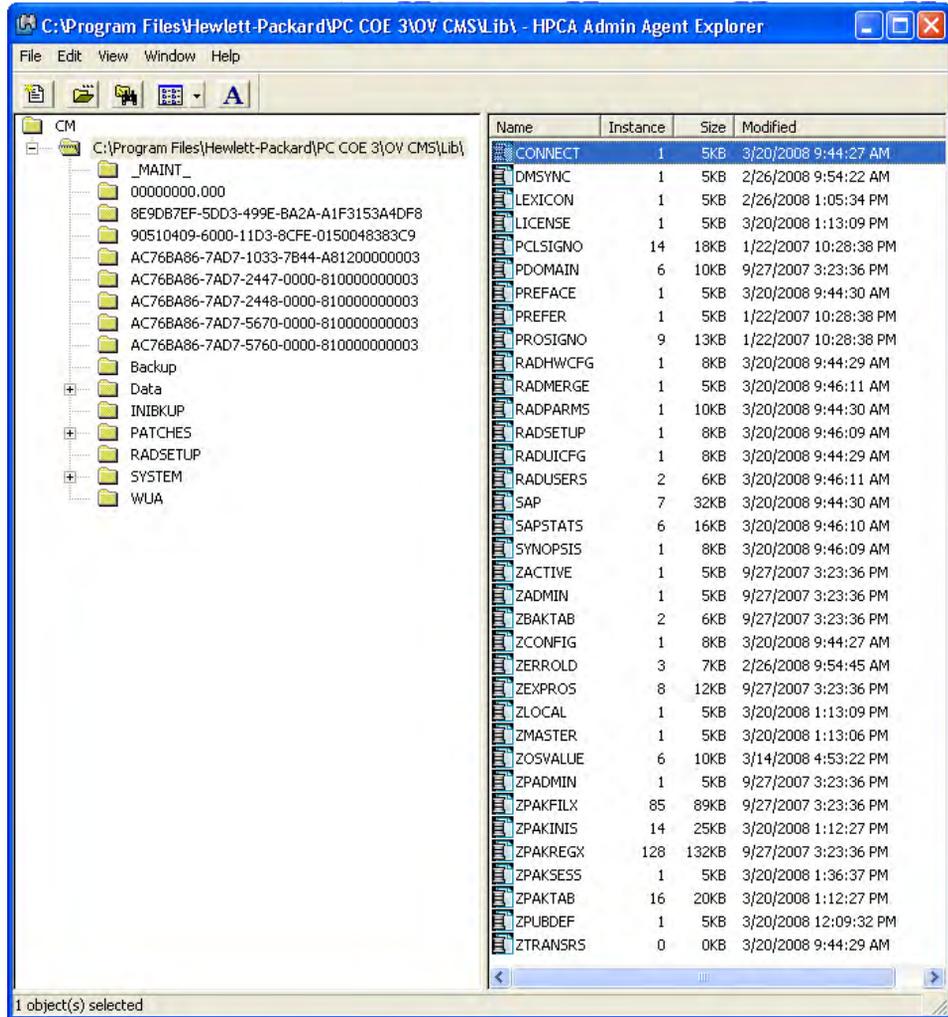
at the command line type **NVDOBJED . EXE**.

The Agent Explorer opens with the object list displayed in the right pane. See [Figure 3](#) on page 40. The object list contains the names of Client Automation objects that you can view and edit.



Objects that begin with "Z" are used by Client Automation. When naming a new object that you create, we recommend that you do not click a name beginning with the letter "Z."

Figure 3 Agent Explorer window



Menus in the Agent Explorer Window

Use the menus in the Agent Explorer window to manage objects, and to manage the appearance of the Agent Explorer window.

File Menu

Use the File menu to open, copy, delete, rename, and create new objects.

Table 5 Agent Explorer File menu

Menu option	Function
New	Click New , type a new object name, and click OK .
Open	Select an object from the object list and click Open . The object view window opens.
Copy	Select an object from the object list, click Copy , and type the name of a new object to which you want to copy the object characteristics, then click OK .
Rename	Select an object from the object list, click Rename , and type the new name over the old name in the list.
Change Directory	Click Change Directory , navigate to the directory you want, then press Enter .
Defined Directories	Click Defined Directories . From the shortcut menu select Admin , Lib , or System .
Exit	Click Exit to close the Agent Explorer.

Edit Menu

Use the Edit menu to navigate objects.

Table 6 Agent Explorer Edit menu

Menu option	Function
Select All	Place the cursor in the right pane, then click Select All .
Byte Convert	Select an object from the object list. Click Byte Convert to convert to Little Endian or Big Endian.
Find in Objects	Click Find in Objects , enter your search criterion, click Find . From the list of objects returned, click the object you want to view to open the object view window.

View Menu

Use the View menu to customize the way the Agent Explorer window looks.

Table 7 Agent Explorer View menu

Menu option	Function
Toolbar	To view the toolbar at the top of the window click Toolbar .
Status Bar	To view the status bar at the bottom of the window click Status .
Full Row Select	To highlight the full row of object information (including the its number of instances, size, and when it was last modified) in the object list when it is selected, click Full Row Select
Show Grid Lines	To view the grid lines in the object list, click Show Grid Lines .
Refresh	To refresh the window, click Refresh .
Object Options	Click Object Options , then select one or more of the following: <ul style="list-style-type: none">• Live Editor Edits contents of disk instead of memory.• Use Recycle Bin Sends deleted objects to the Recycle Bin.• Esc Closes Object Allows you to press Esc to close an object.
View Styles	Click View Styles , then select whether you want to view the objects as icons or in a list.
User Defined Directory	Click User Defined Directory , and type the name of a path to load.
Font	Click Font , and then choose the font you want to use for display in all windows.

Window Menu

Use the Window menu to close all objects.

Table 8 Agent Explorer Window menu

Menu option	Function
Close All	To close all open objects, click Close All .

Help Menu

Use the Help menu to view information about the Agent Explorer and to register it as the default editor for `.edm` files.

Table 9 Agent Explorer Help menu

Menu option	Function
Register as Default Viewer	Sets the Agent Explorer as the default editor for <code>.edm</code> files.
About Agent Explorer	Displays version and copyright information.

Object View Window

The Object View window displays the contents of a Client Automation object for you to view or edit, and contains four drop-down menus: Object, Variable, Instance, and Options.

Object Menu

Use the Object menu to manage, navigate, and print objects.

Table 10 Object menu options

Menu option	Function
Information	Displays the Object Information screen.
Sort	Sorts the heaps of a multi-heap object by the values of variables.
Sort Order	Choose an ascending or descending sort order.
Print	Prints the contents of the object view window.
Export	Exports the object to a <code>.CSV</code> file and puts it in the same location as the object file.

Menu option	Function
Save	Saves the changes you made.
Close	Closes the Object View window.

Variable Menu

Use the Variable menu to add, delete, and rename variables, and manage and navigate through the heaps in an object based on their variable's values.

Table 11 Variable menu options

Menu option	Function
Add	Adds a variable with a user-specified name to an object.
Delete	Deletes a variable from an object.
Edit	Changes the value of a variable.
Modify All	Changes the value of this variable for all heaps in an object.
Filter	Displays all the variables that contain a user-specified character string.
Rename	Renames a variable in an object.
Calculate Row Value	For a selected variable, sums all the decimal only entries for all heaps in the object.
Find	For a user-selected variable, finds the heaps whose values contain a user-specified character string. Available only for multi-heap objects.
Find Next	After you use the Find command to enter a character string to find, this command finds the next occurrence of this character string.
Find Prev	After you use the Find command to enter a character string to find, this command finds the previous occurrence of this character string.
Replace	Find and replace a user-specified character string in a variable's value.

Instance Menu

Use the Instance menu to add, delete, and copy instances in an object, and to navigate through the instances in an object.

An instance is also called a heap. To view each instance in an object, click >> or << at the bottom of the Object View window.

Table 12 Instance menu options

Menu option	Function
Add	Adds an instance with blank values. Go to Variable → Edit to enter the values.
Delete	Deletes the instance you select. Click >> or << to navigate to the instance that you want to delete.
Duplicate heap	Makes a copy of the instance you select, including its values. Click >> or << to navigate to the instance you want to duplicate.
Copy heap to	Copies the instance that you select to the object that you type in the Copy Instance to dialog box.
Go to	Jumps to the instance whose number you type in the New Instance # dialog box.
Home	Jumps to the first instance in an object.
End	Jumps to the last instance in an object.

Options Menu

Use the Options menu to manage the Object View window.

Table 13 Options menu options

Menu option	Function
Resolve Values	Displays variable substitution for a named object and attribute using the ampersand convention &(variable). <ul style="list-style-type: none">• &(DATE) will display the DATE variable contents in that heap of the object.• &(ZERROR.ZERRDATA) will display the ZERRDATA variable contents from the ZERROR object in the current directory.
Show Length	Displays the number of characters allowed for an instance's value.
Split View	Displays the variable's values for two consecutive instances at the same time.
Save Window Position	Saves any changes to the format of the window that you made in the Options menu.
Object Format	Select Little Endian for PC and Big Endian for UNIX and Mac.

Redirecting the CSDB Editor to another Database

You can use the Agent Explorer to redirect the CSDB Editor to another database.

To redirect the CSDB Editor to another database

- 1 Open the Agent Explorer.
- 2 Select the ZMASTER object and right-click to open the Object View Window.
- 3 Change the value of ZIPADDR to the name or IP address of the computer where the other Configuration Server is running.

The CSDB Editor will now be directed to the database that resides on the computer.

Packager

Use the Packager for Windows to create packages using Installation Monitor Mode.

In **Installation Monitor Mode**, the Packager determines what to package by scanning the computer before and after you install the software. It differences the before and after scans to determine what changes were made to the computer after the installation. These differences become the package that you will publish to the CSDB. We recommend this mode for packaging when you do not know all of the components that make up an application.

Set Package Properties

Before you begin to package data, you might want to use the Global Default Properties dialog box to set default properties for the files that will be included in the package. Changes made to the default properties apply to new Packager sessions.

If necessary, you can modify the properties later for any file or folder from the Instance Properties dialog box in Installation Monitor Mode.

You can also use the CSDB Editor to modify the default properties in the base instance. Or, after you promote the package to the CSDB, you can modify the properties for an individual instance in the FILE, REGISTRY, or DESKTOP Classes.

Global Default Properties and **Instance Properties** have three tabs: Agent Management, Data Options, and Agent Behaviors.

To access the Global Defaults Properties dialog box

- 1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator Packager**.
- 2 In the Packager Security Information dialog box, type your User ID and Password in the appropriate text boxes.



The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the New Password and Verify New Password text boxes.

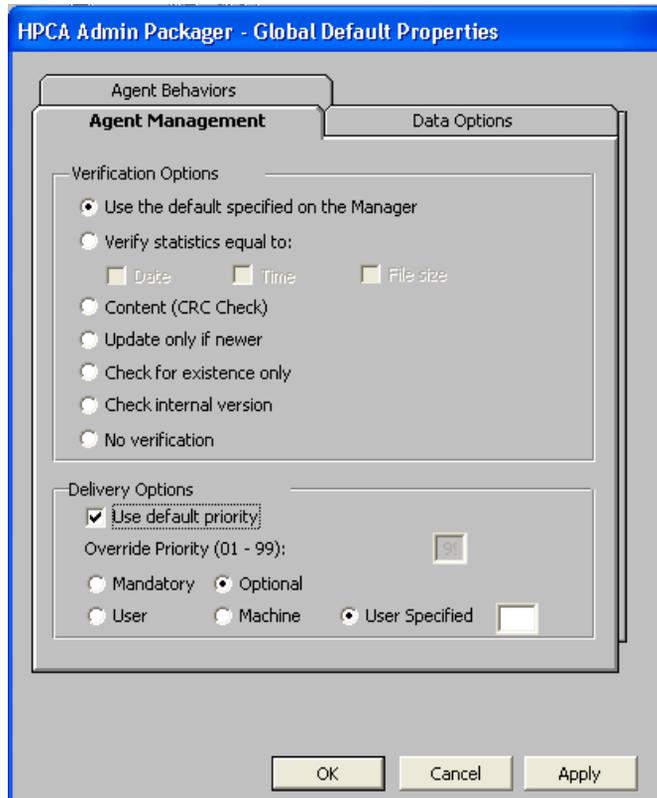
- 3 Click **OK**.

- 4 From the Edit menu, select **Change Global Defaults**. The Packager – Global Default Properties dialog box has three tabs: Agent Management, Data Options, and Agent behaviors.

Agent Management Tab

Use the Agent Management tab to set verification and delivery options for the selected file or folder in the package. After publishing this package to the CSDB, you can still modify these settings using the CSDB Editor.

- Use Verification Options to specify the verification actions that the Client Automation agent will take for this file or folder.
- Use Delivery Options to specify delivery options, such as the order in which files are deployed.



Verification Options

- **Use the default specified on the Manager** (default)
Select this option so that verification options for this file or folder are inherited from the base instance of the FILE Class in the CSDB.

Use the CSDB Editor to look at the ZRSCVRFY attribute of the base instance of the FILE Class to determine what the default verification options are.
- **Verify statistics equal to**
Select this option so that the agent checks the selected statistics (Date, Time, File Size) for the file on the agent computer. The file is deployed from the CSDB (or Proxy Server) if its statistics are different from the file on the agent computer. You can also use the CSDB Editor to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.
- **Content (CRC Check)**
Select the CRC option so that the Client Automation agent checks the statistics of the file on the agent computer. When the file is verified, the *first* the Date, Time, and File Size are checked. To save time during the verify, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Configuration Server (or Proxy Server) will deploy a file if its statistics are different from those on the agent computer. By selecting this option, the Packager will calculate the CRC and put the value in the ZRSCCRC attribute.
- **Update only if newer**
Select this option so that this file is deployed from the CSDB (or Proxy Server) if it has a later date/time stamp than the one on the agent computer. You can also use the CSDB Editor to set this option: ZRSCVRFY=U.
- **Check for existence only**
Select this option so that this file is deployed only if it is not on the agent computer. No action is taken if it already exists on the agent computer, even if its statistics differ from those in the CSDB. You can also use the CSDB Editor to set this option: ZRSCVRFY=E.
- **Check internal version**
Select this option to compare the version of the file being deployed against the version that may already exist on the agent computer. This ensures that the correct file will be updated if more than one version exists. You can also use the CSDB Editor to set this option: ZRSCVRFY=V. This feature applies to the following file types: .DLL, .EXE, .OCX, .SYS, .VBX.

- **No verification**

Select this option so that the file is deployed the first time the application is deployed. No subsequent action is taken. You can also use the CSDB Editor to set this option: ZRSCVRFY=N.

Delivery Options

- **Use default priority**

Select this check box to use the default priority of 50. Priority determines the order of deployment, from highest priority to lowest priority. You can also use CSDB Editor to set this option: ZRSCPRI=50.

- **Override Priority (01-99)**

Type a number from 1 to 99 to override the default priority of 50. 1 is the highest priority and 99 is the lowest. You can also use CSDB Editor to set this option: ZRSCPRI=1.

The following options apply *only* if there is not enough space on the agent computer to install the entire application.

- **Mandatory**

Select this option to indicate that this file is critical to the application.

- If there is not enough space on the agent computer for the entire application, Client Automation will *only* deploy mandatory files.
- If there is not enough space for the mandatory files, then the application is not deployed at all.

You can also use the CSDB Editor to set this option: ZRSCMO=M.

- **Optional** (default)

Select this option to indicate that a file is not critical to the application.

- If there is not enough space on the agent computer for the entire application, Client Automation will *not* deploy optional files.

You can also use the CSDB Editor to set this option: ZRSCMO=O.

The following options apply only to operating systems supporting multiple users with a required sign on, such as Windows NT 4.0, 2000, or Windows Server 2003.

- **User**

Select **User** if you want to indicate that the file should be deployed only to the subscriber logged on when the application is initially deployed. You can also use the CSDB Editor to set this option: ZCONTEXT=U.

- **Machine**

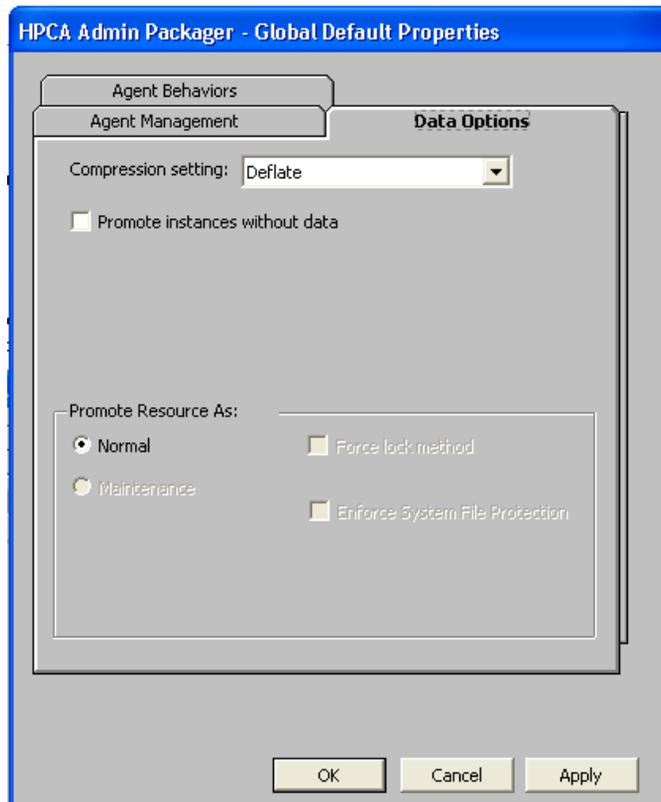
Select **Machine** to indicate that the file should be deployed to all users of

the computer. You can also use the CSDB Editor to set this option:
ZCONTEXT=M.

- **User Specified**
This option is reserved for future use.

Data Options Tab

Use the Data Options tab to specify data compression and other details about the files or folders that you want to distribute.



- **Compression setting**
If necessary, select the compression setting for storing this file in the CSDB. Compression minimizes the time required to transmit data and the amount of disk space required to store it.
 - Select **Deflate** for the most efficient compression, which produces smaller compressed images.

- Select **None** if the file is already compressed. If you are packaging an application that contains a compressed file, do *not* have Packager also compress it, as the file may actually grow in size if it is compressed again.
 - **Promote instances without data**
Select this check box to indicate that this file should not be transferred to the CSDB as part of the package. Only the *instance* representing it is included in the package. The data remain in compressed format in the IDMDATA location on your computer. The default IDMDATA location is `SystemDrive:\Program Files\Hewlett-Packard\CM\Agent\Lib\Data\`.
-  We recommend that you do *not* use this property without a thorough understanding of the extended security in the NT operating system.

Promote Resource As

- **Normal**
Select this option to indicate that this file should be deployed as part of an application.
- **Maintenance**
This feature can be used for versions of Radia prior to the 4.0 release. As of version 4.0, this feature no longer used.
- **Force lock method**
Select this check box to force the use of the **locked file method** for deploying this file. If the file is in use on the agent computer when Client Automation attempts to deploy a new copy of it, the locked file method is normally used to deploy it. If necessary, it is decompressed and stored locally in a folder. The agent connect process forces a restart when it ends and the file is deployed to its correct location during startup.
- **Enforce System File Protection** (Windows 2000 and Windows Server 2003 only)
Select this check box to specify that you want to enforce Windows System File Protection (SFP). Client Automation first attempts to verify the resource using the current verify options, as specified in the ZRSCVRFY attribute located in the FILE Class.
 - If the verification is successful, the resource is in the desired state and no further verification is necessary.

- If the verification fails, Client Automation checks to see if System File Protection protects the file. If the file is protected, it will not be deployed.

Finally, Client Automation checks this attribute to see whether you want to enforce SFP.

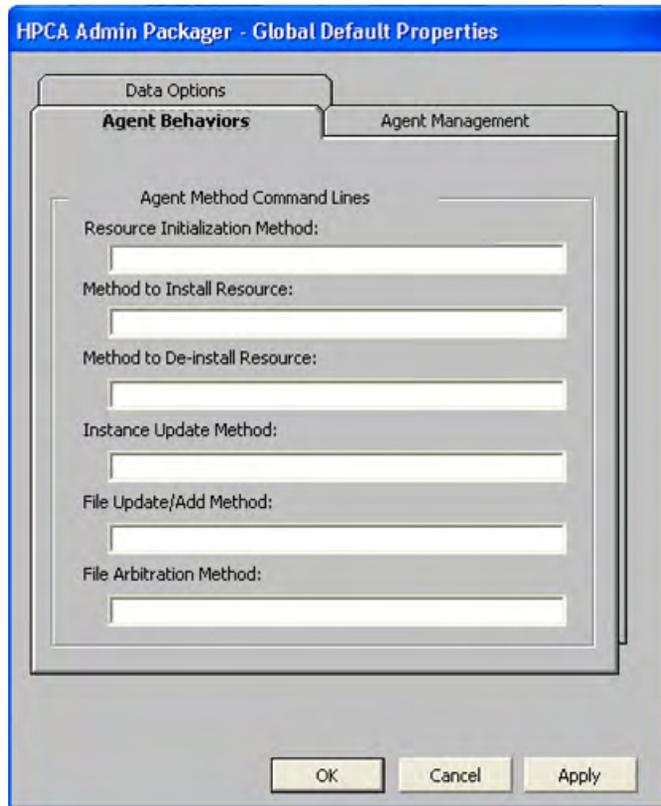
- If you select this check box, the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing.
- If you do not enforce SFP, the file is marked as having passed verification.

A ZERROR instance is created to describe the actions taken against the SFP file.

In the WinZip example, we accepted the default data options.

Agent Behaviors Tab

Use the Agent Behaviors tab to specify methods (programs) that Client Automation executes on the agent computer.



The command lines that you type in this dialog box are stored in attributes in the FILE Class instances in the SOFTWARE Domain.

Agent Method Command Lines

- **Resource Initialization Method** (Attribute in database: ZINIT)
Type the method to run when the file is stored on the agent computer.
- **Method to Install Resource** (Attribute in database: ZCREATE)
Type the method to run after the file has been stored on the agent computer. This is used if some processing is required to enable it to be used on the agent computer.



To allow deployment of applications that require user interaction on Vista devices use the "runasuser" method modifier. During publishing, add the modifier "runasuser" to the Method to Install Resource text box.

For example:

```
runasuser setup.exe
```

Note: The method modifier "runasuser" cannot be used along with modifier "hide" as these modifiers are mutually exclusive.

- **Method to De-install Resource** (Attribute in database: ZDELETE)
Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared DLL file, is not supposed to be deleted from the agent computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for the **Method to De-install Resource**.
- **Instance Update Method** (Attribute in database: ZUPDATE)
Type the method to run when the instance is modified on the computer, after the file has been deployed.
- **File Update/Add Method** (Attribute in database: ZFILEUPD)
Type the method to run when the file is new or has been updated. This method executes just before it is deployed to the agent computer.
- **File Arbitration Method** (Attribute in database: ARBITRAT)
Type the method to run if a file is about to be replaced. This method compares the version information of the file or folder on the agent computer and the one that is going to replace it, and then determines which to keep.



You must add the ARBITRAT method to the CSDB to use this field. See the Technical Note *About the File Arbitration Method* on the HP Software web site for more information.

Component Select Mode

The Packager no longer supports Component Select mode. However, you can enable Component Select mode in the Packager to package links or registry keys. To enable component selection mode in the Packager, add a variable to ZMASTER called PKGCOMP and set the value to 'Y'. Component Selection mode will then be enabled for the Packager.

Installation Monitor Mode

You use five phases to create a package in Installation Monitor Mode:

- 1 **Define** the application's hardware, software, and operating system requirements.
- 2 **Analyze** (scan) the hardware and software on the computer.

- 3 **Install** the software that you want to package.
- 4 **Analyze** (scan) the hardware and software on the computer again. The difference between the pre- and post-scans represents the software you installed.
- 5 **Publish** the package to the CSDB.

Recommendations

The scans performed in Installation Monitor Mode are highly effective. Therefore, we will highlight some of the subtleties that you might want to consider. If you are already familiar with systems management software, some of this material may be redundant.

Before you package an application in Installation Monitor Mode:

- Set up a **clean computer**.
A clean computer has the same operating system as your target audience and has the Administrator installed. Do not make any other changes to the computer.
- Use the same operating system and language version as the target computers for all packaging activities.



There are distinct differences between the national language implementations of operating systems. These differences may include different registry key names and directory locations, which can create a significant operational gap between the expected behavior of the packaged application and the actual operation of the application.

For example, if you are installing the French version of Office 2000 for Windows on computers using the French language version of Windows 2000 or Windows Server 2003, package the application on a computer with the French language version of the target operating system.

- Package on a computer with a static IP address.

If your packaging computer uses Dynamic Host Configuration Protocol (DHCP) to obtain a dynamic IP address, each time the computer is restarted you will receive a different IP address. If you are using Installation Monitor Mode, and the software you are packaging causes the computer to restart, the IP address in the registry will change. Therefore, we recommend packaging on a computer with a static IP address.

- If you are packaging Visual Basic applications, create a Client Automation package that includes the Visual Basic 6 support files.

When installed with the Administrator, the CSDB Editor is a Visual Basic 6 application. Therefore, the following Visual Basic 6 support files are installed:

ASYCFILT.DLL	COMCAT.DLL
COMCTL32.OCX	COMDLG32.OCX
MSVBVM60.DLL	OLEAUT32.DLL
OLEPRO32.DLL	STDOLE2.TLB

If you use Installation Monitor Mode to package a Visual Basic 5 application on a computer that has CSDB Editor installed, the support files listed above will not be included in the package. These files exist on the computer, even before the pre-installation scan is done. Therefore, the post-installation scan will not recognize these files as being added when the software is installed.

To distribute the necessary support files with your Visual Basic applications, use the Component Select Mode in the Publisher to create a package that includes only the Visual Basic 6 support files. Then, include this package in any application package for Visual Basic 6.

The alternative is to package Visual Basic 6 applications on a computer that does not have the CSDB Editor installed. To install the Administrator without the CSDB Editor, put the CD-ROM in the drive and type the following on the command line:

```
CDDrive:\HPCAЕ-Admin72.msi -ne
```

Using Installation Monitor Mode to Create a Package

This section explains how to use Installation Monitor Mode to create a package, and also provides detailed information about each of the screens you will encounter.

Task 1 Log on to Packager

- 1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator Packager**.
- 2 In the Packager Security Information dialog box, type your User ID and Password in the appropriate text boxes.

- ▶ The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the New Password and Verify New Password text boxes.

- 3 Click **OK**.

Task 2 Complete the Open Packaging Session Window

Use the Open Packaging Session window to select the packaging mode and enter information to identify the packaging session.

At the end of a **packaging session**, you will have a package—a unit of distributable software or data—that you will connect to a service. You will set up policies to distribute the software or data to the targeted subscribers.

- ▶ Review the Global Defaults prior to beginning any Packaging Session. See [Set Package Properties](#) beginning on page 47.
- ▶ In previous versions of Packager, Import Mode was available for applications installed or maintained using Windows Installer. For information on packaging these applications, see the *HP Configuration Management Extensions for Windows Installer User Guide (Extensions for WI Guide)*. This is a separately licensed product.

- 1 In the Session Type area, **Installation Monitor Mode** is the only option available. Component Select Mode is now only available in the Publisher. See [Publisher](#) on page 72.
- 2 In the What to Open area, select **New Session** to begin a new packaging session. You must complete the Session ID and Description fields.
or
Select **Existing Session** to resume a previous session.
- 3 In the Session ID field, type a unique 1 to 6 character name for the session. If you do not complete this packaging session, you will need to remember this ID to return to it later.
- 4 In the Description field, type a description for the session.
- 5 Click **Next**. The Package Properties window opens.

Task 3 Enter Package Properties

Use the Package Properties window to name the package and include additional descriptive information.

- 1 In the Package Name field type a name for the package. This is the name for the PACKAGE Class instance in the CSDB, and should conform to your naming conventions. The name cannot contain any spaces.
 - ▶ You might want to establish a naming convention to ensure that identifiers are unique. Packager uses this identifier to construct data objects and file names.
 - ▶ If you enter a name of an existing package you will receive an error message. Go back and enter a different name for this new package, or go back and open the existing package.
- 2 In the Domain field select the domain in which you want to store the instance. This is normally the SOFTWARE Domain unless you customized the CSDB with proprietary domains.
- 3 In the Description field type a description of the package.
- 4 In the Release field, type the release number of the software.
 - ▶ The current session information is stored in *SystemDrive*:
\\Program Files\Hewlett-Packard\CM\Agent\LIB
\\ZPAKSESS.EDM on your computer. Use the Agent Explorer to view this object.

If you decide to close this packaging session and do not plan to complete it, you can delete this object. You can also use the CSDB Editor to delete the PACKAGE Instance from the CSDB. Locate the object in the SOFTWARE Domain, Packages Class.
- 5 Click **Next** to go to the System Configuration window.

Task 4 Set the Required System Configuration

Use the System Configuration window to limit the distribution of the package to computers that meet specific requirements. Distribution is based on the computer's operating system, installed RAM, and processor.

- 1 Select the Target Operating System check box to activate the Operating System list box. Then, select one or more operating systems required for this package.

- To select multiple, consecutive operating systems, press and hold down the **Shift** key on your keyboard and click the items.
- To select multiple, non-consecutive operating systems, press and hold down the **Ctrl** key on your keyboard and click the items.

If you do not select an option, the package will be available to subscribers running all operating systems.

- 2 Select the Minimum Megabytes of Memory Required check box to activate the Megabytes options. Then, select the minimum RAM required for this package.

If you do not select an option, the package will be available to subscribers with any amount of available memory.

- 3 Select the Minimum Processor Required check box to activate the Processor options. Then, select the minimum processor that is required for this package.

If you do not select an option the package will be available to subscribers with any processor.

If none of the above three options is selected, the package will be available to all eligible subscribers.

- 4 Click **Next** to go to the Availability window.

Task 5 Set Date and Time Constraints

Use the Availability window to specify the date and/or time when the package will be available for deployment. The date and time are based on the system clock of the computer running the Configuration Server.

If you do not specify date and time constraints, the package will be available as soon as it is promoted to the CSDB and configured for distribution.

- 1 Select the Before check box to prevent distribution of the package *before* the specified date and time. Use the Month, Day, Year, Hour, and Min text boxes to specify the date and time.
- 2 Select the After check box to prevent distribution of the package *after* the specified date and time. Use the Month, Day, Year, Hour, and Min text boxes to specify the date and time.
- 3 Click **Next** to go to the What to Scan — Set Packaging Options window.

Task 6 Identify What to Scan

Use the What to Scan: Set Packaging Options window to identify areas where the software might make changes to the target computer. You can choose to scan the desktop, registry, file system, or to compare file content.

Before you install the application, the Packager scans the selected areas and takes a **snapshot** that records their current state. After you install the application, the Packager takes another snapshot. Client Automation compares these before and after snapshots, and the differences make up the package.

If you do not know how the installation of the application will affect the target computers, scan all areas to ensure that all changes made by the application are recorded.

The more information that you have about the software that you are packaging, the easier it will be to package and deploy. One benefit of knowing the application is that you can save time during scanning. Below are some examples of this.

- If you know that the application does not make any registry changes, you can clear the Registry check box so that Packager does not scan the registry.
- If you are packaging some favorites to Internet Explorer and you know that Internet Explorer stores favorites in individual .URL files, you can scan only the file system to pick up the changes.
- If you are packaging several bookmarks to Netscape Navigator and you know that Netscape Navigator stores bookmarks in a single .HTM file, you can include this file in the File Contents to be Scanned list, select the File Content Comparison check box, and Packager picks up the changes to the bookmark file.



Client Automation does *not* support the packaging of files from the IDMSYS, IDMLIB, or IDMLOG directories.

- 1 To identify which areas of the computer you want to scan, select one or more of the following check boxes:
 - File System
 - Desktop (Icons, Shortcuts, Groups)
 - Registry
 - File Content Comparison

- 2 If you select File Content Comparison, you must include the file name in the File Contents to be Scanned list box. To do this, type the full path and file name in the File Name text box, or click **Browse** to navigate to and select it. Then click **Add**.
- 3 Click **Next**. If, in step 1, you chose to scan the File System, the Set Drives to Scan window opens. If not, the Pre-Installation window opens (go to Task 8.)

Task 7 Identify What to Scan in the File System

Use the What to Scan — Directories screen to select the areas of the file system that you want to scan. All of the drives (local and network) connected to your computer are considered the **file system**. Many factors, including the computer's hardware and software, affect the amount of time it takes to scan the system.



This window opens *only* if you selected the File System check box in the Set Packaging Options: What to Scan window.

- 1 Select *all* the folders or drives that will be affected by the installation.
 -  We recommend that you do *not* scan network drives. These can take longer due to the overhead transmission across the network. Also, the mapping of network drives may vary from one user to another.
- 2 Click **Add** to move your selections into the Directories to be Scanned list.
- 3 Click **Next** to go to the Pre-Installation Scan window.

Task 8 Begin the Pre-Installation Scan

Use the Pre-Installation Scan window to begin the scan.

- 1 Click **Begin Scan**. A check mark appears next to each area as the scan is finished.
- 2 When the scan is done, a message box opens.
- 3 Click **OK**.

Task 9 Install the Software

After the pre-installation scan is complete, you must install the software.

- 1 In the Pre-Installation Scan window, click **Next**. A message opens.

- 2 Click **OK**. The Packager minimizes and Packager icon appears in the system tray.
- 3 Install the application using its native installation.
 -  Do *not* do anything other than install the application. If you do, you risk making changes to the file system, registry, desktop, and so on. Client Automation will detect these (unrelated) changes during the post-installation scan and include them in the package.
 -  If the application that you are installing requires that you restart your computer, the Packager icon in the system tray will be available when you log on again.
- 4 After the application is installed, double-click the Packager icon in the system tray. A message asks if the installation was successful.
- 5 Click **Yes**. The Packager logon window opens.
- 6 Type your User ID and Password, and then click **OK**. The Post Installation Scan window opens.

Task 10 Begin the Post-Installation Scan

Use the Post-Installation Scan window to do the final scan.

-  The comparison of the registry scans may take a long time, and there may be very little disk activity. Although it may seem as though no progress is being made during this step, be patient. The registry scan comparison will complete normally. Use a clean computer to minimize the time it takes to complete this comparison.

To begin the post-installation scan

- 1 Click **Begin Scan**. When the scan is done, a message box opens.
- 2 Click **OK**. A list of how many items were found that were different from the pre-installation scan appears.
 -  After differencing the before and after scans, the Packager creates three registry (.EDR) files in the REGISTRY class. These files are published to the Configuration Server.
- 3 Click **Next** to go to the Summary window.

Task 11 Review the Modifications to the File System

Use the summary window, to review the changes that were made when you installed the software on your computer.

- 1 Click the **Files** tab to review the files that were added, deleted, or updated.
- 2 Click the **Desktop** tab to review the program groups, links, and shortcuts that were added, deleted, or updated.
- 3 Click the **Registry** tab to review changes detected in the registry.

Detailed information about the Files, Desktop, and Registry tabs follows, beginning with [View File Properties and Locations](#) below.

Task 12 View File Properties and Locations

Use the Files tab in the Summary window to see the files and folders that will be included in the package.

To view the selected items

- 1 Right-click the topmost **Selected** branch in the tree and select **Expand All**. Check marks indicate that the properties for the file have been specified and the Packager is ready to publish them.
- 2 Select an item in the tree view to see its properties in the list view. Some of the properties such as Verify, Priority, and Mandatory/Optional, are initially set according to the selections in the Packager – Global Default Properties dialog box, as described starting on page 47.

To filter the displayed items

If the package contains many items, you can use filtering to limit the type of items that you want to see in the window.

- 1 Click the filter button  to filter the items in the tree view.
- 2 In the Set Filter dialog box, type the item type as a string, such as `.DLL`. If the filter string is found anywhere in the item's name, it is considered a match.
- 3 Click **OK**. Only those items that meet the filter requirements are displayed.

To remove a filter

- 1 Click the filter button  to open the Set Filter dialog box.
- 2 Click **Clear**.
- 3 Click **OK**. All files are displayed.
- 4 Click **Next** to continue.

Task 13 Set File Properties and Locations

You can modify the properties of the files and folders displayed on the Files tab, or you can exclude them from the package.

To edit a file's properties

- 1 Right-click a file. A shortcut menu opens with the following three options.
 - Set properties
 - Active
 - Inactive
- 2 To exclude this file from the package, click **Inactive**.
- 3 To include this file in the package, click **Active**. This is the default setting.
- 4 To modify the properties of the file, select **Set Properties**. The Instance Properties dialog box opens.
- 5 Use the Instance Properties dialog box as described in [Set Package Properties](#) on page 47.

To edit a folder's properties

- 1 Right-click a folder. A shortcut menu opens with the following three options.
 - Set properties
 - Active
 - Inactive
- 2 To exclude this folder from the package, click **Inactive**. All the files in that folder will also be inactive.

- 3 To include this folder in the package, click **Active**. This is the default setting. All the files in that folder will also be active.
- 4 To modify the properties of the folder, select **Set Properties**. You must also indicate whether you want the changes to affect the Directory Only or Directory and Files.
- 5 Select Directory and Files. The Instance Properties dialog box opens.

Use this dialog box to modify the attributes of the files in the package. The properties of a file are stored in its instance in the CSDB. The default values of these properties are set in the Global Default Properties dialog boxes. For more information see [Set Package Properties](#) on page 47.

Task 14 Updated Files

In addition to the files that were added, you might notice Updated files.

Updated files are control files whose ASCII text has been modified, such as configuration files. You can set the properties for a standard ASCII text file as described in [Set File Properties and Locations](#) on page 65.

If a control file, such as `WIN.INI`, has been updated, the file name of the desktop object is listed, followed by the name of the section in the `.INI` file that has been modified. Below the section name, you will see the changes to individual lines. Each line is listed separately so that you can edit them as necessary.

To edit a line in a control file

Right-click the line that you want to modify in the tree view.

- Click **Active** to include the change to the text file in the package.
or
- Click **Inactive** if you do *not* want to include the change to the text file in the package.
or
- Click **Override Value** if want to modify the change to the text file. The Override Text File Data dialog box opens.

To use the Override Value feature

- 1 Click **Override Value**.
- 2 Type the New Value for the line in the text box.

- 3 Click **OK**.

To restore the text line

- 1 Right-click the line that you want to modify in the tree view.
- 2 Click **Override Value**.
- 3 Click **Reset**.
- 4 Click **OK** to save your changes and close the dialog box.

Task 15 Set Desktop Properties and Locations

Use the Desktop tab in the Set Properties and Locations window to see the selected program groups, links, and shortcuts in the package, and to modify their properties.

To view the selected program groups, links, and shortcuts

- Right-click **Selected Links**, and select **Expand All**.

To set properties for program groups, links, and shortcuts

- Right-click the link in the tree view. A menu opens with options listed below.



When you select a folder or link, Client Automation applies the changes to all links within that folder or within the link's branch in the tree view.

— **Just In Time (App Mgr Only)**

The HPCA Application Manager (App Mgr) lets you designate applications as **self-maintaining**. This means that each time the subscriber launches the application, a connection is made to the Configuration Server to verify the application and, if necessary, update or repair it.

Note that the use of Just In Time with Machine/User applications can only verify or repair either the Machine or User portion of the service. If the Just In Time icon is using context=M, then startdir should be set to SYSTEM, and the user portion of the service will not be verified/repared. If the Just In Time icon is using context=U, then startdir should be \$USER, and only the user portion of the service will be verified/repared.

— **Active**

Select **Active** to include this link or folder in the package.

- **Inactive**
Select **Inactive** to exclude this link or folder from the package.
- **Properties**
Select **Properties** to open the Verify/Type submenu.

Verify Menu Commands

- **Full Verify**
Select this command to deploy the link every time a subscriber connects to the Configuration Server, even if it already exists on the agent computer. This repairs broken or modified links.
- **No Verify**
Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.
- **Exists**
Select this command to deploy the link only if it does *not* exist on the agent computer during initial deployment and on each subsequent connection. The link is deployed again only if it was deleted. This allows users to modify the link.

Type Menu Commands

The following options apply only to operating systems that support multiple users, such as Windows 2000, XP, or Windows Server 2003.

- **User**
Select this command to deploy this link only to the subscriber logged on when the application is initially deployed.
- **Machine**
Select this command to deploy this link to all users of the computer.

Task 16 Set Registry Properties and Values

Use the Registry tab in the Set Properties and Locations window to see the changes that will be made to the registry when the application is deployed. You can also modify the values in registry keys and set deployment options.

To view the selected registry

- Right-click **Selected Registry Keys**, and select **Expand All**.

To edit values in a registry key

- 1 Select the registry key in the tree view.
- 2 Double-click the registry key in the list view.
- 3 Edit the data in the New Value text box.
- 4 Click **OK** to save your changes and close the dialog box.

To restore the registry key to its original value

- 1 Double-click the registry key in the list view.
- 2 Click **Reset**.
- 3 Click **OK** to save your changes and close the dialog box.

To set properties for registry keys

- Right-click the link in the tree view. A shortcut menu opens.



When you right-click a registry key, Client Automation applies the changes to all registry keys within the key's branch in the tree view.

- **Active**

Select this command to include registry changes in the package.

- **Inactive**

Select this command to prevent registry changes from being included in the package.

- **Properties – Deploy / Verify**

- **Unconditional/Enforced**

Select this command to deploy the registry key, even if it already exists on the agent computer. If the registry key is modified, the next time the subscriber connects to the Configuration Server, this registry key is reset to the packaged value. Use this setting for registry keys that must not change in order for the application to function properly.

- **Unconditional/Exist** (default)

Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset.

- **Exist/Exist**

Select this command to deploy the registry key only if it *does not*

exist on the agent computer. This allows you to modify the registry key and your changes will not be overwritten because the key will be deployed again only if it is deleted.

Task 17 Publish the Package

You now need to create instances for the package in the CSDB.

- After you have set all file properties and locations, click **Next**. The Packager creates the object to be published. This is the package that will be published to the CSDB.



After you publish the package, the session's objects are deleted and you *cannot* resume the session. Therefore, you may want to save the session so that you can reopen it later. For example, you might find problems while testing the deployment of the package, and want to make corrections. Here's how to save the session:

Copy the `LIB` directory (and its subdirectories) from `SystemDrive:\Program Files\Hewlett-Packard\CM\Agent\LIB` to a backup location. The `LIB` directory contains the objects that were built by the Packager during your packaging session.

By saving the `LIB` directory, promoting the package, and then restoring the `LIB` directory, you can alter the settings in the Packager's windows, and publish the package again. Use the toolbar buttons to jump directly to the various windows in the Packager.

If you find a problem while testing the deployment of the promoted package, this allows you to fix it.



If you need to change or modify your selections, click **Prev** until you reach the appropriate window. When you are satisfied with the package, click **Next** until you arrive back at this window.

You can also use the buttons in the toolbar to return to a previous screen.

To view the files to be promoted

- Right-click **Files to be Promoted**, and select **Expand All**.

To publish the package

- 1 Right-click **Files to be Promoted**, and select **Expand All** to view the files that will be promoted.

- 2 Click **Promote**.
- 3 Click **OK** when you receive the prompt that the package has been promoted successfully.
- 4 Click **Finish** to close Packager.
- 5 Click **Yes** to confirm that you want to close the Packager.

Next, use the CSDB Editor to create a service. For more information, see [Creating Services with the New Application Wizard](#) on page 170.

Publisher

The Administrator Publisher (Publisher) is a tool you use to package data and publish it to the CSDB in a form that Client Automation can easily manage.

▶ When you publish an application that is targeted for Vista deployment, if you choose to install it under system account, then you must make sure that the application will deploy without any user interaction on the target Vista machine.

Overview

The Publisher supports four different types of publishing. [Table 14](#) below lists these types and where to find information about each one. This book describes only two types: Component Select and Windows Installer files.

Table 14 Types of files published by Publisher

Publication Type	Where to find information
Component Select	Publishing in Component Select Mode on page 73
Hardware Configuration	<i>HP Client Automation OS Manager Hardware Configuration Guide</i>
Operating System images	<i>HP Client Automation OS Manager System Administrator Guide</i>
Windows Installer files	Publishing Windows Installer Files on page 88

Batch Mode

The Publisher also allows for the publication of files in batch mode. This lets you use a configuration file to publish multiple `.msi` files to the CSDB during one publishing session. See [Publishing Multiple .msi Files \(Batch Mode\)](#), on page 103 for more information.

Standard Publishing Steps

First you must start and log on to the Publisher.

To start the Publisher

- 1 On the device where you installed the Publisher, use the **Start** menu and go to:

Start→All Programs→HP Client Automation Administrator→ Client Automation Administrator Publisher

- 2 Use the Administrator password to log on to the Publisher. By default, the user name is **rad_mast** and the password is **secret**.

Each of the publishing modes that are discussed in this guide – Component Select and Windows Installer – consists of four basic tasks:

Task 1 - Select window — Select the files that you want to package.

Task 2 - Edit window — Edit the file properties.

Task 3 - Configure window — Enter package and service information.

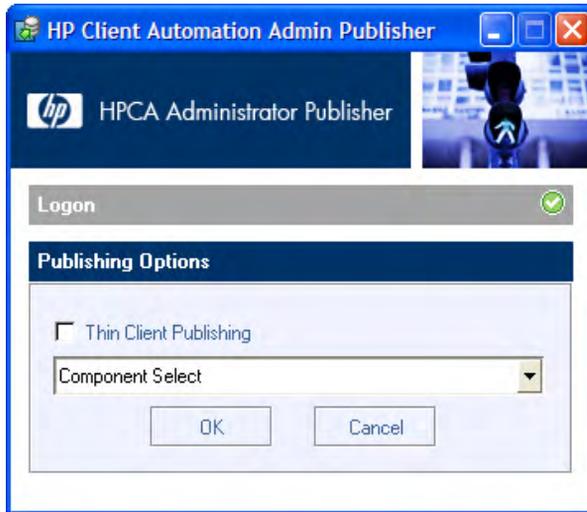
Task 4 - Publish window — Publish the package to the CSDB.

Publishing in Component Select Mode

As of the 5.1 release of the Publisher contained a new feature: Component Select Mode. Prior to this release this mode of publishing was available in the Packager. It is now unavailable in the Packager, and is available only in the Publisher.

To publish in Component Select Mode

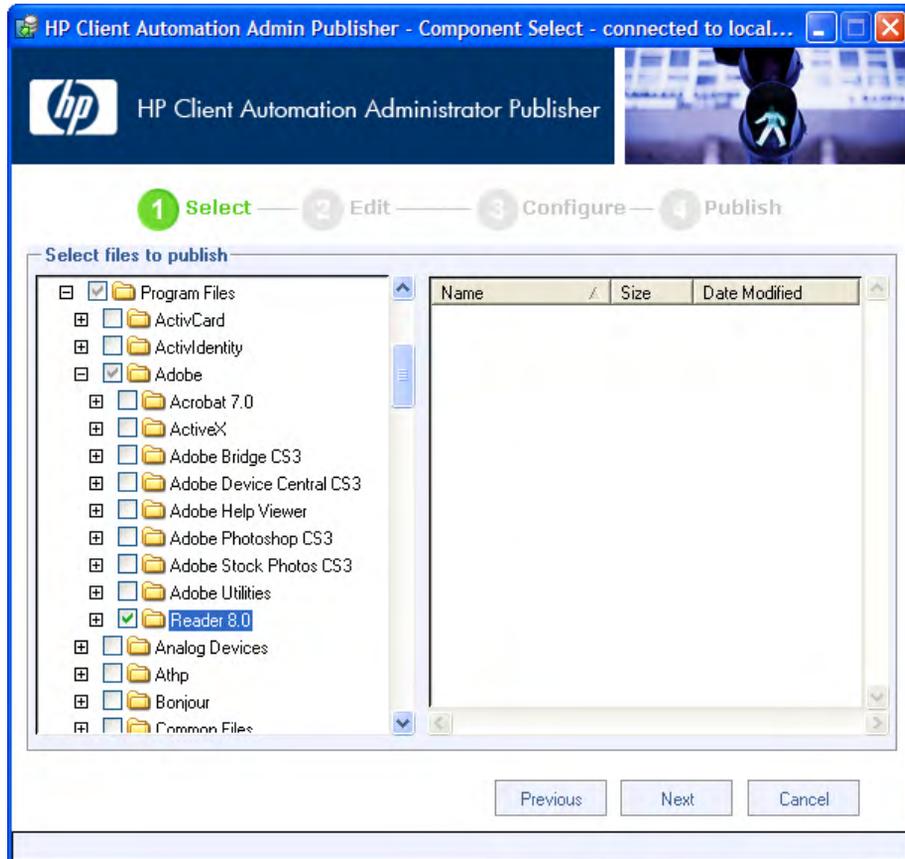
- 1 Start the Publisher. The Publishing Options widow opens.



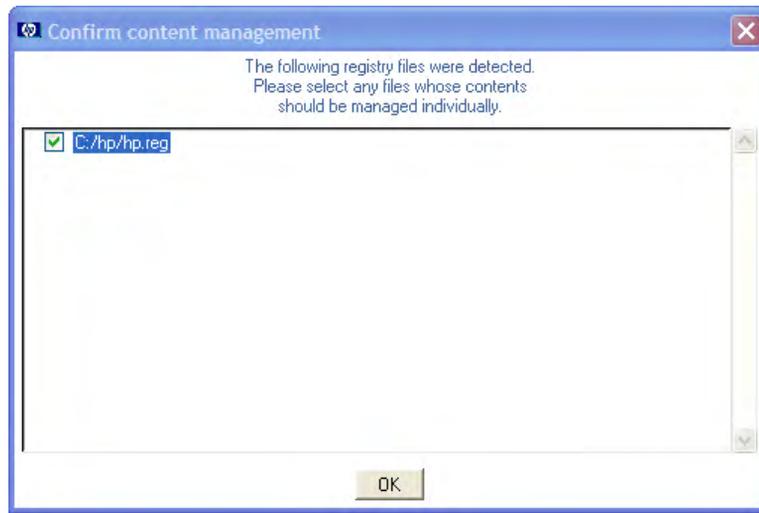
- 2 In the Publishing Options window:
 - If you are publishing for thin clients, select **Thin Client Publishing**.
 - From the drop-down list, select **Component Select**.
- 3 Click **OK**. The Select window opens.

Task 1 Select Files to Publish

Use the Select window to select the files, folders, registry keys, and links that you want to include in the package.



- 1 Navigate through the file system (left pane) and select all the items that you want to include in the package.
- 2 Click **Next**.
 - If you selected one or more registry files, a Confirm content management dialog box opens. Select the registry files whose content you want to manage individually, and click **OK**.



- If you selected one or more link files, a Confirm content management dialog box opens. Select the link files whose content you want to manage individually, and click **OK**.

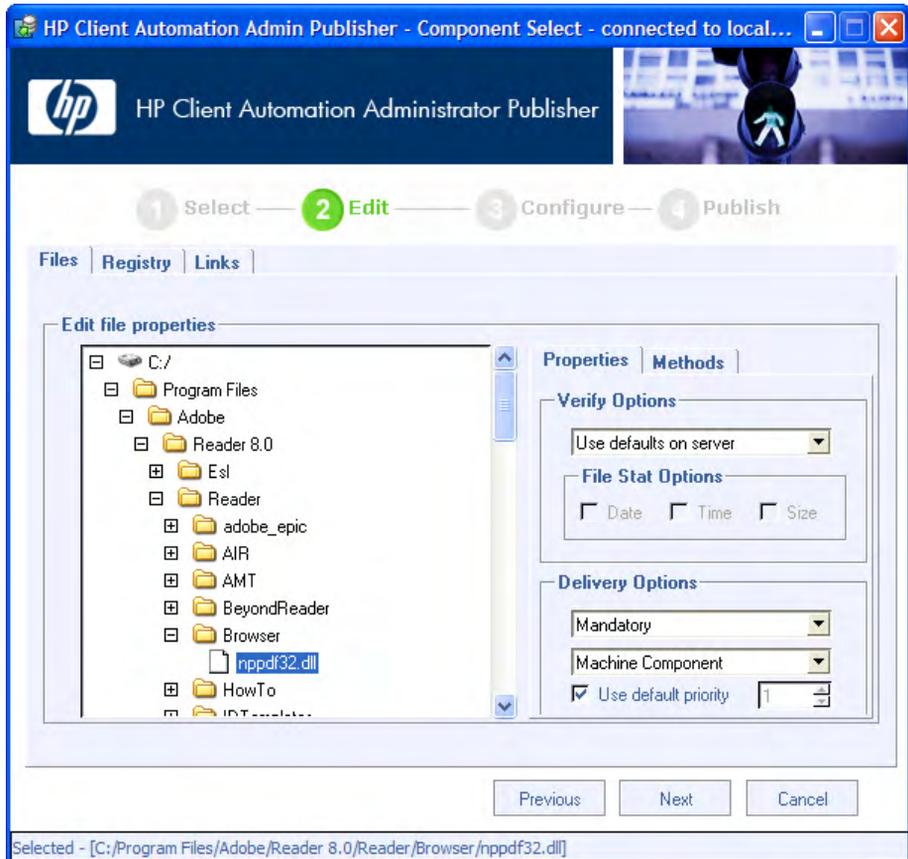
The Edit window opens (go to Task 2, [Edit Properties and Methods](#)).

Task 2 [Edit Properties and Methods](#)

Use the Edit window to edit properties and methods.

[Editing File and Folder Properties](#)

- 1 Select the Files tab.
- 2 All files and folders that will be included in the package are listed in left pane.



- 3 Select any files or folders whose properties you want to edit.
- 4 Click the **Properties** tab to edit the file or folder's verification and delivery options.

Verify Options

- **Use defaults on server** (default)
Select this option to inherit verification options from the FILE, REGISTRY, or DESKTOP Class' base instance in the CSDB.
- **File statistics**
Select this option so that the Client Automation agent checks the selected statistics (Date, Time, or File Size) for the files or folders on the computer. Select the Date, Time, or File Size using the check boxes below the Verify options drop-down list. The files or directories are deployed from the CSDB if the statistics of the files or folders on the computer are different from the statistics for these files or folders

in the database. You can also use CSDB Editor to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.

— **Content check using CRC**

Select this option so the HPCA Application Manager checks the statistics of the file on the agent computer. When the file is verified, *first* the Date, Time, and File Size are checked. To save time during the verify, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Configuration Server (or Proxy Server) will deploy a file if its statistics are different from those on the agent computer. By selecting this option, the Publisher will calculate the CRC and put the value in the ZRSCCRC attribute. ZRSCVRFY is set to Y.



Content CRC checking is a time consuming process and should be used sparingly.

— **Update if newer**

Select this option so that these files or folders are deployed if the files or folders in the CSDB have a later date/time stamp than those on the agent's computer. You can also use the CSDB Editor to set this option: ZRSCVRFY=U.

— **Existence only**

Select this option so that these files or folders are deployed if they are not on the subscriber's computer. No action is taken if the files or folders already exist on the agent's computer, even if the statistics differ from those in the CSDB. You can also use the CSDB Editor to set this option: ZRSCVRFY=E.

— **None**

Select this option so that the files are deployed the first time the application is deployed. No subsequent action is taken. You can also use the CSDB Editor to set this option: ZRSCVRFY=N.

Delivery Options

The following options apply only if there is not enough space on the subscriber's computer to install the entire application. Select if the application will be Mandatory or Optional.

— **Mandatory**

Select this option to indicate that these files or folders are critical to the application. If there is not enough space on the agent's computer for the entire application, Client Automation will deploy only mandatory files. If there is not enough space for the mandatory files,

then the application is not deployed at all. You can also use the CSDB Editor to set this option: ZRSCMO=M.

— **Optional** (default)

Select this option to indicate that files or folders are not critical to the application. If there is not enough space on the agent's computer for the entire application, Client Automation will not deploy optional files. You can also use the CSDB Editor to set this option: ZRSCMO=O.

— **Use default priority**

Select the Use default priority check box (default) to use the default priority of 50. Priority determines the order of deployment, from highest priority to lowest priority. You can also use the CSDB Editor to set this option: ZRSCPRI=50.

If you clear the check box, you can enter a number from 1 to 99 to override the default priority of 50. 1 is the highest priority and 99 is the lowest. You can also use the CSDB Editor to set this option: ZRSCPRI=1.

- 5 When you are finished editing file properties, click the **Methods** tab to edit the methods. The command lines that you type in the dialog boxes on the Methods tab are stored in attributes in the FILE Class instances in the SOFTWARE Domain.

— **Resource Initialization Method** (Variable in database: ZINIT)

Type the method to run when the files or folders are stored on the agent computer.

— **Method to Install Resource** (Variable in database: ZCREATE)

Type the method to run after the file or folder is stored on the computer. Use this if some processing is required to enable the file to be used on the agent computer.



To allow deployment of applications that require user interaction on Vista devices use the "runasuser" method modifier. During publishing, add the modifier "runasuser" to the Method to Install Resource text box.

For example:

```
runasuser setup.exe
```

Note: The method modifier "runasuser" cannot be used along with modifier "hide" as these modifiers are mutually exclusive.

- **Method to Uninstall Resource** (Variable in database: ZDELETE)
Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared object file, should not be deleted from the agent computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for Method to De-install Resource.
- **Instance Update Method** (Variable in database: ZUPDATE)
Type the method to run when the instance is modified on the agent computer, after the file has been deployed.
- **File Update/Add Method** (Variable in database: ZFILEUPD)
Type the method to run when the file is new or has been updated. The method executes just before the file is deployed to the agent computer.
- **File Arbitration Method** (Variable in database: ARBITRAT)
Type the method to run if files or folders are about to be replaced. This method compares the version information of the files or folders that exist with the files or folders that are going to replace them, and then determines which to keep.



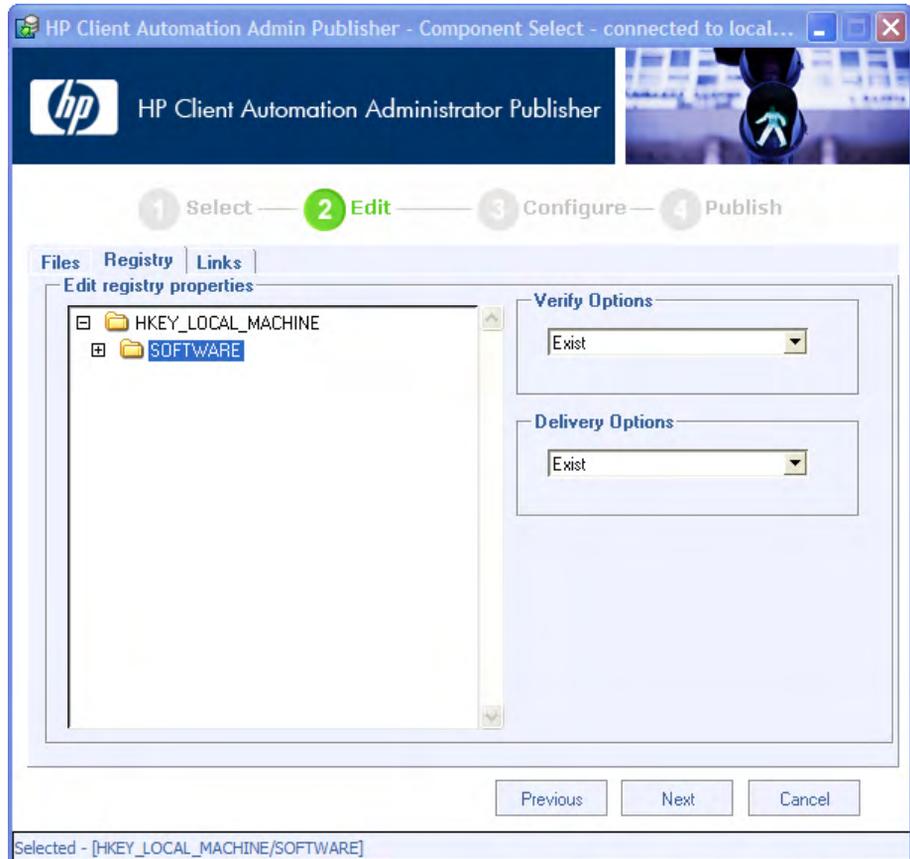
You must add the ARBITRAT method to the CSDB to use this text box. See the technical note about the File Arbitration Method on the HP Software web site.

- 6 When you are finished editing file and method properties, click **Next** to go to the Configure window (go to [Enter Package and Service Information](#) on page 83).

Editing Registry File Properties

In this window you can edit the verify and delivery options for the registry keys you selected.

- 1 Select the Registry tab. All registry keys that will be included in the package are listed in left pane.



- 2 Select any registry keys whose properties you want to edit.
- 3 In the Verify and Delivery Options boxes you may select the following combination of verify and delivery options:
 - **Enforced/Unconditional**
Select this command to deploy the registry key, even if it already exists on the agent computer. If the registry key is modified, the next time the subscriber connects to the Configuration Server, this registry key is reset to the packaged value. Use this setting for registry keys that must not change in order for the application to function properly.
 - **Exist/Unconditional/**
Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset.

— **Exist/Exist**

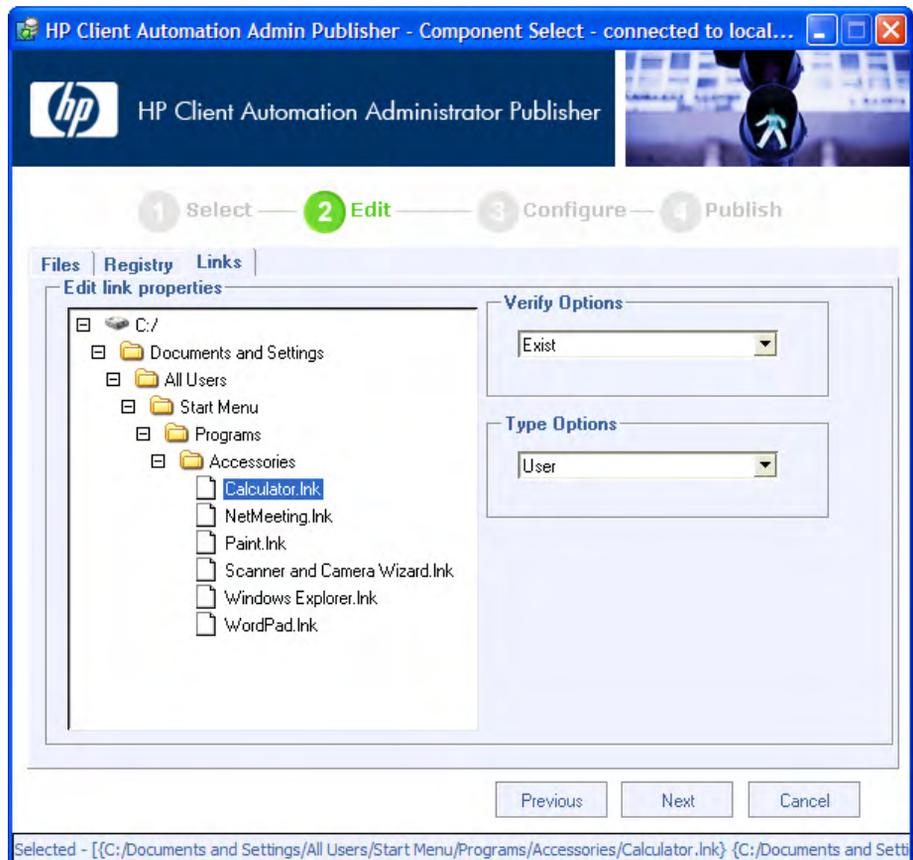
Select this command to deploy the registry key only if it *does not exist* on the agent computer. This allows you to modify the registry key and your changes will not be overwritten because the key will be deployed again only if it is deleted.



You may *not* select a verify value of Enforced in combination with a delivery value of Exist. This is not a valid combination. If you make these selections, the values will revert to a verify value of Enforced, and a delivery value of Unconditional.

Editing Link Properties

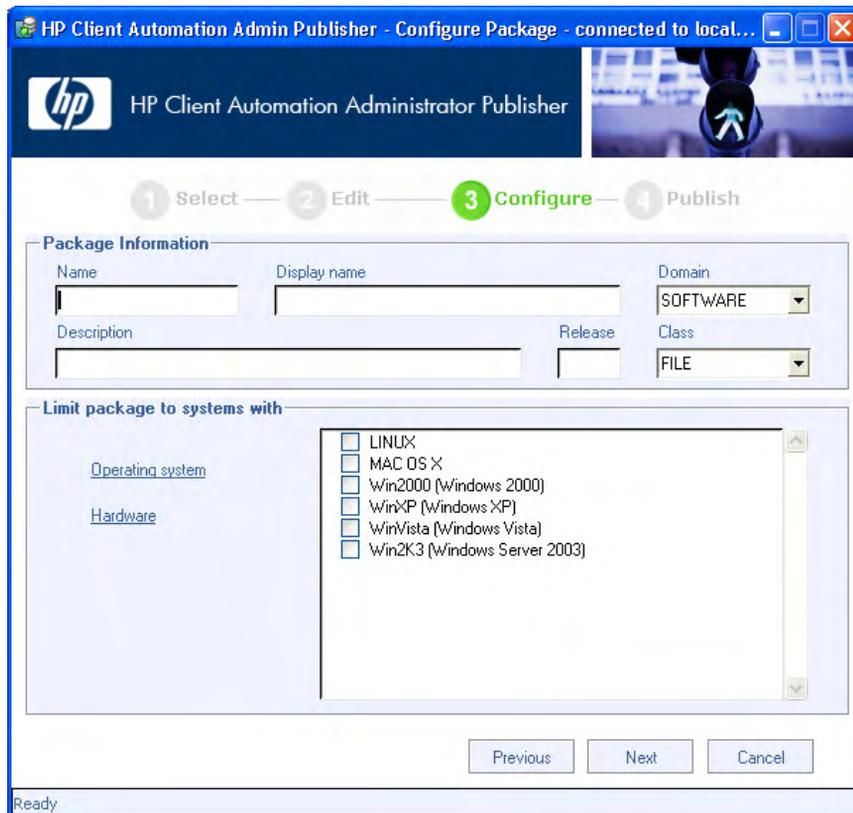
- 1 Select the Link tab.
- 2 All links that will be included in the package are listed in left pane.



- 3 Select any links whose properties you want to edit.
- 4 In the Verify Options box, from the drop-down list, select **Full**, **None**, or **Exist**.
 - **Full**
Select this command to deploy the link every time a subscriber connects to the Configuration Server, even if it already exists on the agent computer. This repairs broken or modified links.
 - **None**
Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.
 - **Exist**
Select this command to deploy the link during initial deployment and on each subsequent connection only if it does *not* exist on the agent computer. The link is deployed again only if it was deleted. This allows users to modify the link.
- 5 In the Delivery Options box, from the drop down list, select **User** or **Machine**.
 - **User**
Select User Component if you want to indicate that the file will be deployed only to the agent that is logged on when the application is initially deployed. You can also use the CSDB Editor to set this option.
 - **Machine**
Select Machine Component to indicate that the file will be deployed to all users of the computer. You can also use the CSDB Editor to set this option.
- 6 When you are finished editing file and method properties, click **Next** to go to the Configure window (go to [Enter Package and Service Information](#) below).

Task 3 Enter Package and Service Information

Use the Configure windows to enter information about the package and the service. Here you name the package and include additional descriptive information and set package deployment limitations based on hardware and operating system settings. You also enter descriptive information about the service, and information about how it will be managed after it is deployed.



- 1 In the Package Information area enter the following information:
 - a In the Name field, type a name for the package. This is the name for the PACKAGE Class instance in the CSDB and should conform to your naming conventions. Note that the name cannot contain any spaces.
 - You may want to establish a naming convention to ensure that identifiers are unique. Publisher uses this identifier to construct data objects and file names. See [Appendix B, Naming Conventions](#) for more information.
 - b In the Display name field, enter a display name for the package. This is the friendly name used in the CSDB Editor.
 - c From the Domain drop-down list, select the domain in which to store the instance. This is normally the SOFTWARE Domain unless you customized the CSDB with proprietary domains. As shipped, the default domains are ADMIN, AUDIT, CLIENT, PATCH, POLICY, PRDMAINT, SOFTWARE, and SYSTEM.

- d In the Description field, type a description for the package.
 - e In the Release field, type the release number of the software.
 - f From the Class drop-down list, select the class in which to store the instance.
- 2 In the Limit package to systems with area, enter the following information. If none of these options is selected, the package will be available to all eligible subscribers.
 - a **Operating system**
Select the operating systems to which this package applies.
 - b **Hardware**
To limit distribution based on minimum RAM or processor speed, select the check box next to the appropriate configuration option and enter the minimum requirement in the text box provided.
 - 3 Click **Next** to go to the Service Information window

HP Client Automation Admin Publisher - Configure Service - connected to local...

HP Client Automation Administrator Publisher

1 Select — 2 Edit — 3 **Configure** — 4 Publish

Service Information

Create new Use existing No service

Name: _____ Display name: _____

Vendor: _____ Web URL: _____

Description: _____ Author: _____

Assignment type

Mandatory Optional

Management type

Automatic Manual

Report on the following events

Use Base

Install	<input checked="" type="checkbox"/> Success	<input checked="" type="checkbox"/> Failure
Remove	<input checked="" type="checkbox"/> Success	<input checked="" type="checkbox"/> Failure
Update	<input checked="" type="checkbox"/> Success	<input checked="" type="checkbox"/> Failure
Verify	<input type="checkbox"/> Success	<input checked="" type="checkbox"/> Failure
Repair	<input type="checkbox"/> Success	<input checked="" type="checkbox"/> Failure

Previous Next Cancel

Ready

- 4 Select whether you want to create a new service (Create new), use an existing service (Use existing), or skip creating a service (No service) at this time.
 -  If you want to create only a package, select **No service**. This is useful if, for example, you have a single service, but want to create multiple packages and later connect them to the existing service using the CSDB Editor.
- 5 Enter a Name, Display name, Vendor, Web URL, Description, and Author.
 -  If you are using an existing service, make sure you enter the service name correctly.
- 6 In the Assignment type section, select whether the service is mandatory or optional. **Mandatory** is the default. This will distribute this service to all available subscribers. Optional services are only available if you are using the HPCA Application Self-service Manager.
- 7 In the Management type section, indicate how the application will be managed after it is deployed.

Management type and Assignment type correspond to the ZSVCMO service attribute value. This field can contain two values, depending on what you select for each type. The table below displays the possible attribute values.

Table 15 ZSVCMO Variables

Assignment Type	Management Type	ZSVCMO Value	Explanation
Mandatory	Automatic	M	Service is deployed to all subscribers and managed by Client Automation.
Mandatory	Manual	MO	Service is deployed to all subscribers and is not managed by Client Automation after deployment.
Optional	Automatic	OM	Service is deployed only to those subscribers that accept it and is managed by Client Automation.

Assignment Type	Management Type	ZSVCMO Value	Explanation
Optional	Manual	0	Service is deployed only to those subscribers that accept it and is not managed by Client Automation after deployment.

- 8 In the Report on the following events section, select each check box next to the events you would like to record. If you would like to use the default application reporting events available in the Base Instance, select **Use Base**.
- 9 Click **Next** to open the Publish window.

Task 4 Publish the Package

Use the Publish window to view the package and service information and then to publish it.



If you need to change or modify your selections, click **Previous** until you reach the appropriate window. When you are satisfied with the package, click **Next** until you arrive back at this window.

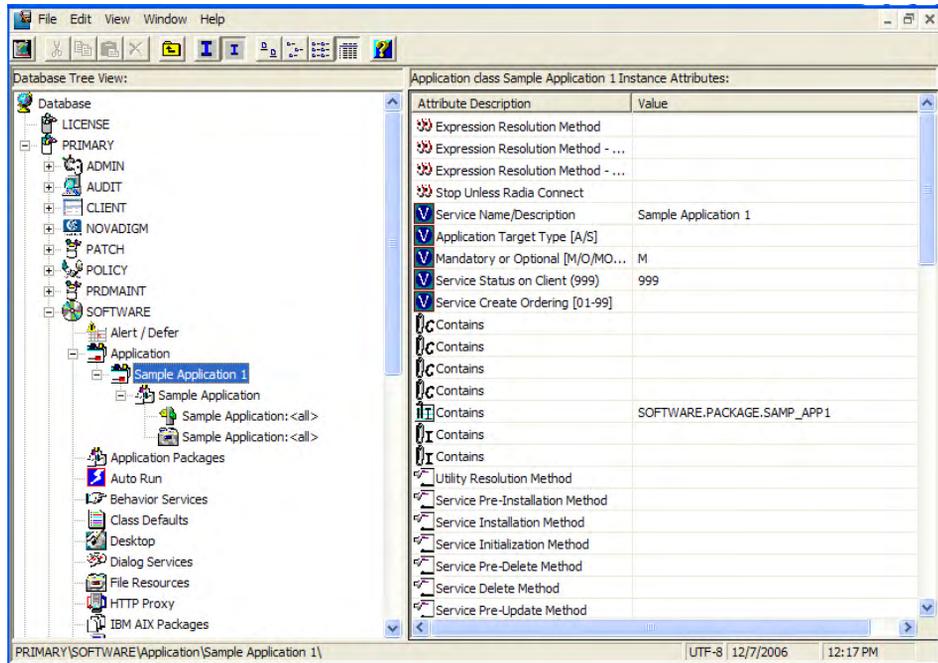
You can also use the buttons in the toolbar to return to a previous screen.

To publish the package

- 1 Click **Publish**. The package is published to the database.
- 2 When the status bar at the bottom of the window displays the message "Process completed successfully," click **Finish**.
- 3 Click **Yes** to confirm that you want to exit.

You can now view the published service in the CSDB Editor.

Figure 4 Sample Application



Publishing Windows Installer Files

Windows Installer uses MSI files to distribute software packages to your operating system. The Publisher uses the files to create a package and service that are then published to the CSDB. After they are added to the CSDB, the software packages are ready for distribution to your environment.

VMware ThinApp Isolated Applications

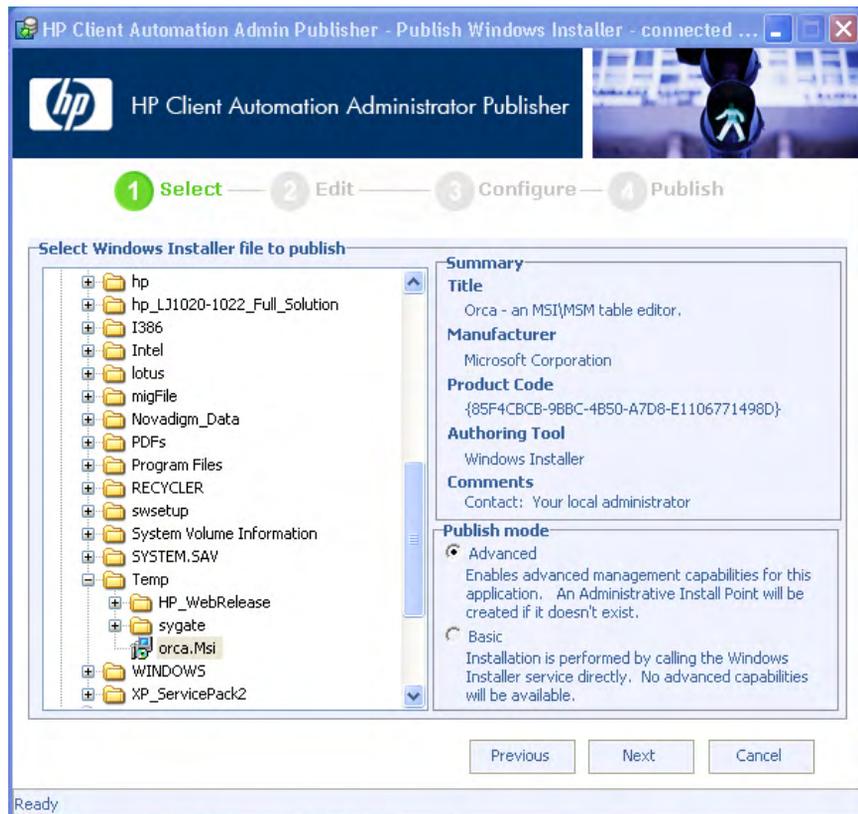
The Administrator Publisher has been updated to recognize the existence of a VMware ThinApp Isolated Application packaged inside an MSI. If detected, Advanced Publishing Mode (see [Advanced Publishing Mode](#) on page 91) will be disabled and the associated service will be marked as containing a VMware ThinApp. The administrator will be asked if he wants to enable auto-updates of the ThinApp by using VMware's AppSync facility. If selected, this option will also be indicated inside the associated service for future processing. This change is the first step in tighter integration with virtual applications, and will be leveraged in future improvements in this area.

To publish Windows Installer files

- 1 Start the Publisher.
- 2 From the Type of data to publish drop-down list, select **Windows Installer**.
- 3 Click **OK**. The Select window opens.

Task 1 Select Files to Publish

Use the Select window to select all the files that you want to include in the package.



- 1 In the left pane select the Windows Installer file that you want to publish. The right pane displays any information that is available for the .msi file you select.
- 2 In the Publish mode section, select either **Basic** or **Advanced**.
- 3 Click **Next** to go to the Edit window.

Task 2 Edit File Properties and Methods

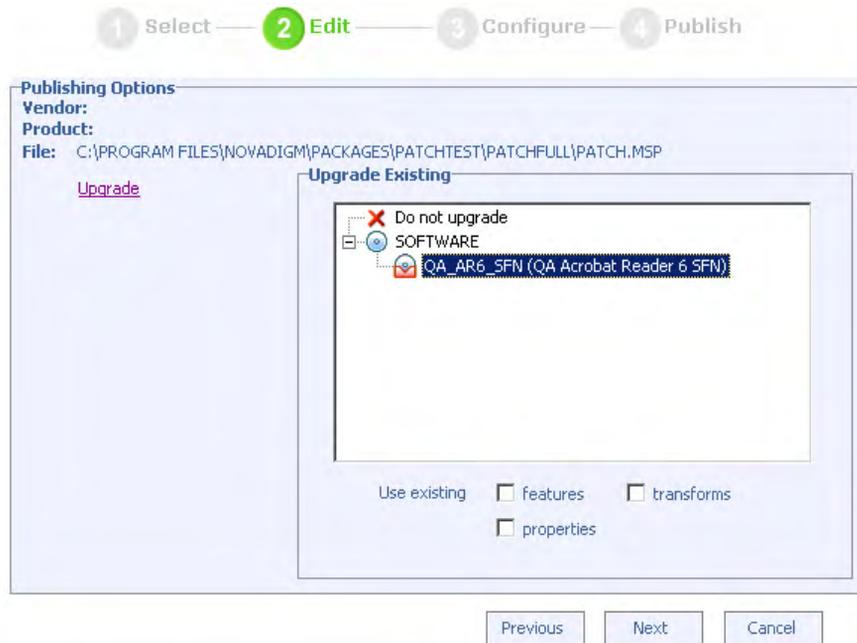
This task differs depending on whether you:

- selected a Microsoft patch file (.msp) in Task 1, Step 1. If so, go to [Upgrading an Existing Package](#) below.
- selected the Advanced publishing mode in Task 1, Step 2. If so, go to [Advanced Publishing Mode](#) on page 91.
- selected the Basic publishing mode in Task 1, Step 2. If so go to [Basic Publishing Mode](#) on page 98.

Upgrading an Existing Package using an .MSP File

If you selected a Microsoft patch file (.msp) to publish, the Edit window will have only one option: Upgrade. The Upgrade Existing window shows all available packages in the CSDB that can be upgraded with the selected file.

- 1 From the Upgrade Existing window, select the corresponding package that you want to upgrade.



- 2 Click any of the check boxes in the Use existing section at the bottom of the window to use the existing package's features, properties, or transforms.

You can also upgrade an existing package using another Windows Installer file. When publishing this type of file (.msi), the Advanced Publishing Mode presents you with the Upgrade option. See [Advanced Publishing Mode](#) below for more information.



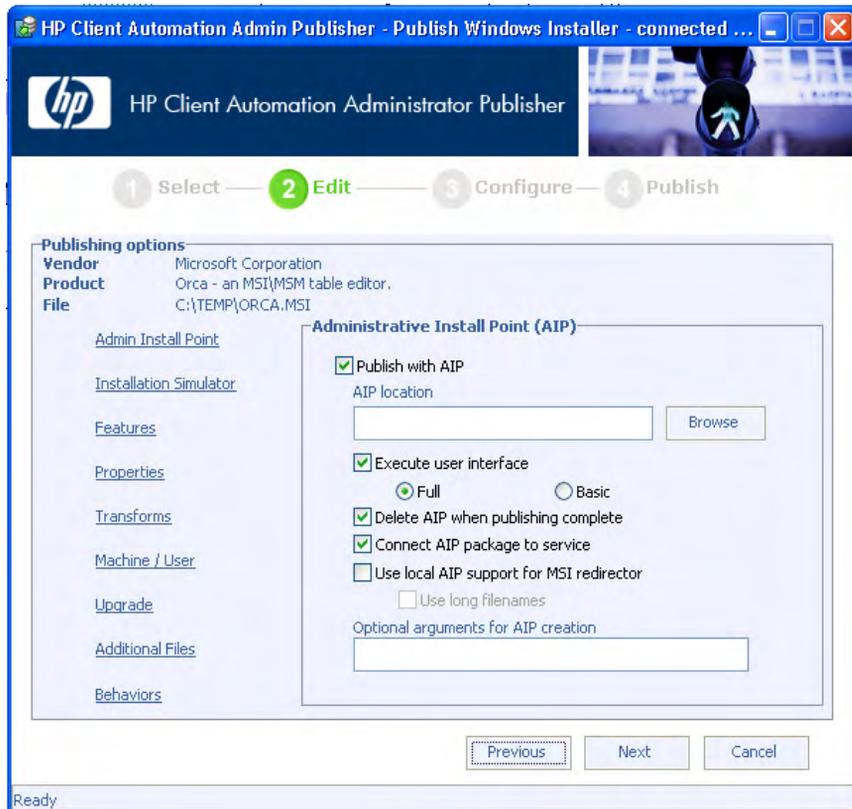
The Publisher will create a new package for the upgrade. After the new upgrade package is published to the CSDB, connect it to the existing package using the CSDB Editor.

Advanced Publishing Mode

The advanced publishing mode allows you to edit many of the publishing options used to create the package.

To publish in Advanced Publishing Mode

- 1 If the Administrative Install Point (AIP) window is not open, click the **Admin Install Point** option on the left. The Administrative Install Point (AIP) window opens.



Creating the Windows Installer AIP

Previously, the Windows Installer publishing process required the manual creation of an AIP outside of the publishing process. Starting with the Publisher 4.0, this is no longer required. Now, you can supply the `.msi` file in its original state without any modifications and the Publisher then automatically creates the AIP in the short file name format.

2 In this window make the following selections to create an AIP:

Table 16 Creating an AIP

Publish with AIP	Select this check box to publish using an AIP.
AIP location	Type in or click Browse to designate the AIP location.
Execute user interface	Select to execute either a full or basic user interface after package installation.

Publish with AIP	Select this check box to publish using an AIP.
Delete AIP when publishing complete	Select to delete the AIP after the package is published to the CSDB.
Connect AIP package to service	Select to connect the AIP package to the associated service.
Use local AIP support for MSI redirector	Select to create a temporary local AIP to support Client Automation advanced data management and caching during the installation. Enable this feature for installations that do not inherently support HTTP-based MSI installs. After selected, choose the Use long filenames check box to use long file names.
Optional arguments for AIP creation	Type any additional arguments to include during the AIP creation.

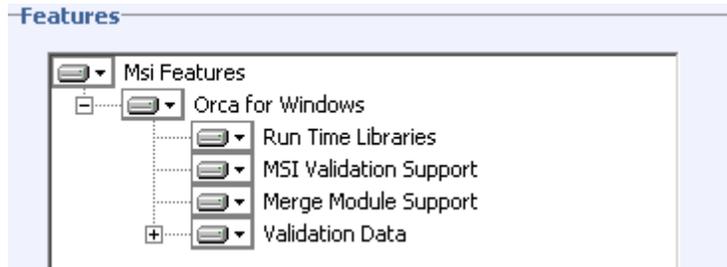


In order to connect the AIP package to the service, select the **Connect AIP package to service** check box. When this check box is selected, the AIP package will be connected to the service and will have its ZSTOP000 set to 1. This prevents the AIP package from being delivered to the desktop, but its connection to the service will guarantee its travel during Proxy Server operations.

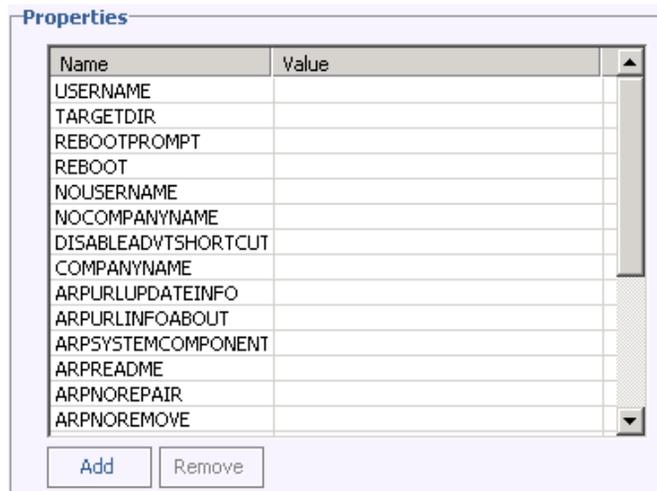
- 3 Click **Installation Simulator**, then click **Run User Interface**. This will simulate an installation, but will not actually install the software.



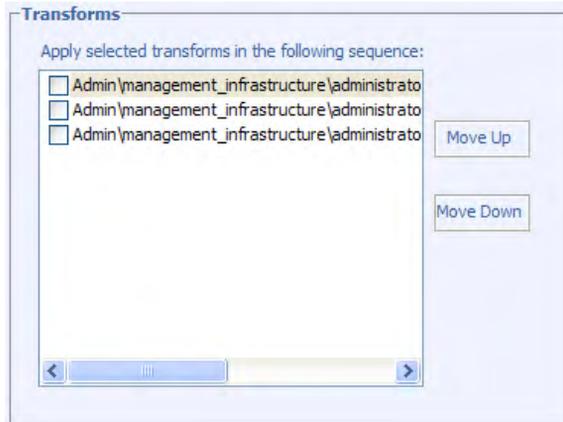
- 4 Click **Features**. The Features window opens. Use this window to configure the available MSI features.



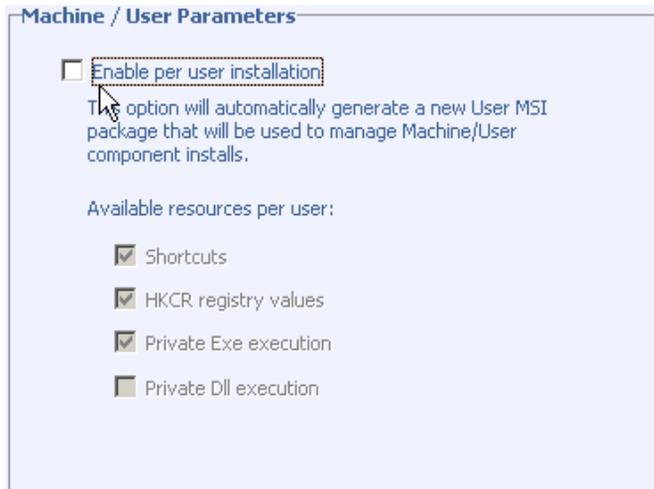
- 5 Right-click each feature and select if you want it to be hidden or disabled.
- 6 Click **Properties**.



- 7 In the Properties window, view and modify the `msi` file properties. Some Windows Installer files may require additional command line parameters to deploy correctly. For example, an application may require a custom property to pass a serial number during installation. Use the Properties window to include any additional parameters.
 - Click **Add** to add a new property.
 - Click **Remove** to delete an existing property.
 - To modify a property **Name** or **Value**, click the item you want to change and enter the new value.
- 8 Click **Transforms**. Use this window to reorder the application of any transform files associated with the Windows Installer file.



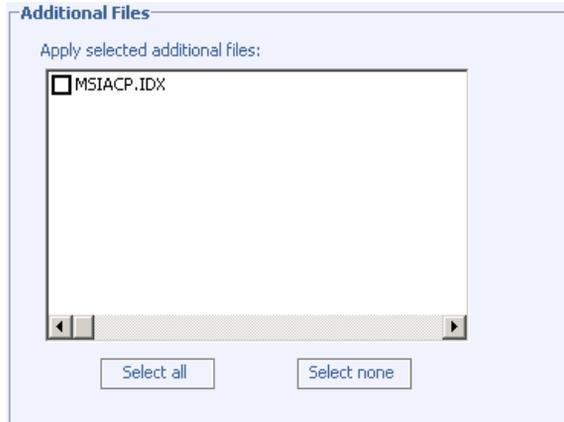
- 9 Click **Machine / User**. Use this window to define Machine/User parameters.



- **Enable per user installation** – Select this check box to automatically generate a new user MSI package to be used to manage Machine/User component installations.
 - **Available resources per user** – Select which resources should be made available per user.
- 10 Click **Upgrade**. This window displays existing MSI packages in the CSDB that are available for upgrade using the current Windows Installer patch (.MSP). Select to use existing features properties or transforms by clicking the appropriate check boxes.



- 11 Click **Additional Files**. Select any additional files that you want to include as part of the AIP.



- 12 Click **Behaviors**. This window contains options for using a local MSI file cache, enabling local repair, saving Internet cache, controlling MSI features, and determining salvage status.

Behaviors

Use local cache for MSI files

Cache location Disk percent

Use local repair

Preserve internet cache

Salvage

Managed applications

Unmanaged applications

User controls MSI features

- **Use local cache for MSI files** – Select to determine the location and size of your cache. Enter the cache location in the Cache location text box and the upper limit of the percentage of disk space to be used in the Disk percent for cache text box.

When you select Use local cache for MSI files, during the application installation, the cache directory is created in the specified location. If the used disk space on the target machine becomes greater than the cache percentage limit, caching is automatically turned off.

- **Use local repair** – Select to enable local repair. Choosing this option retains the MSI cache on the local agent computer, and allows repairs to be made locally, instead of going back to the CSDB for the missing or required data.
- **Preserve internet cache** – Select to avoid having the Internet cache purged after an HTTP installation.
- **Salvage** – Allows you to take advantage of existing files on the target computer. When a package is installed, if any portion of that package is discovered, the installation will use the existing files instead of going back to the CSDB for the redundant data. You have the option to enable the Salvage feature for HPCA-managed and unmanaged applications, select the appropriate check boxes (Managed applications, Unmanaged applications) to determine which type of application should have this feature enabled. Both types of applications can be selected.

- **User controls MSI features** – Select to allow the end user to control which MSI features are installed during the installation. Selecting this option displays the feature editor during the application installation.

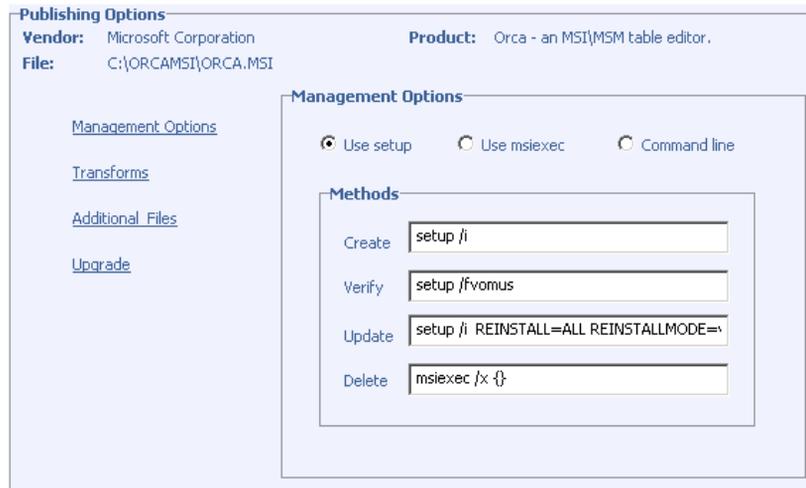
13 Click **Next** to go to the Configure window.

Basic Publishing Mode

The basic publishing mode offers four options, which are available in the Publishing Options window.

To publish in Basic Publishing Mode

- 1 After you select the publishing mode in the Select window, click **Next**. The Edit window opens. This window contains four options: Click any link to configure that aspect of the basic publishing mode.



- **Management Options**
Select which management option you would like to apply. The management option you select defines how the application will run after it is deployed.
 - Use setup
 - Use msixexec
 - Command line

- **Transforms**
Reorder the application of any transform files associated with the Windows Installer file.
 - **Additional Files**
Apply any available additional files.
 - **Upgrade**
Displays existing MSI packages within the CSDB that are available for upgrade using the current Windows Installer package. Select to use existing feature properties or transforms by clicking the appropriate check boxes.
- 2 When you are finished editing the publishing options, click **Next** to go to the Configure window.

Task 3 Enter Package and Service Information

Use the Configure windows to enter information about the package and the service. See [Enter Package and Service Information](#) on page 83 for a detailed description of these windows.

- 1 In the Package information section, enter the package information.
- 2 Use the Limit package to systems with section to limit the package to any specific operating system or hardware. Click any link to display the configurable options.
- 3 Click **Next**. The Service Information window opens.
- 4 Enter service information in this window
- 5 Click **Next** to go to the Publish window.

Task 4 Publish the Package

- 1 Review the Summary section to verify the package and service information you provided during the previous steps. When you are satisfied, click **Publish**.
- 2 When the publishing process is finished, click **Finish** to exit the Publisher. Use the CSDB Editor to view your operating system package or service.

The service is now ready for distribution to your enterprise.

Command Line Publishing

The Publisher supports a non-graphical publishing mode that substitutes any values entered during the Publisher user interface with values predefined in a configuration file. The configuration file allows you to predefine any package and service values and store them in a text file. The Publisher can then be run from the command line with the parameter `-cfg` designating which configuration file to use during the publishing session.

A sample command line to run the Publisher with a configuration file called `test.cfg` would look like the following:

```
nvdtk.exe pubport.tkd -cfg "c:\program files\test.cfg" -  
user USER -password PASSWORD
```



Make sure to use a fully qualified file name when designating the configuration file.

Table 17 below describes the parameters of the configuration file.

Table 17 Configuration file parameters

Parameter	Description
package	Defines the PACKAGE Class instance name.
pkgname	Specifies the friendly name of the PACKAGE Class instance (NAME attribute).
pkgdesc	Specifies a description of the PACKAGE Class instance (DESCRIPT attribute).
service	Defines the name of the ZSERVICE Class instance that will be optionally created.
svcname	Specifies the friendly name of the ZSERVICE instance (NAME attribute). The ZSVCNAME attribute is also set from this value.
svcdesc	Specifies a description of the ZSERVICE Class instance (DESCRIPT attribute).
addtosvc	Controls whether or not a service instance is created for this session. 1 = create service 0 = don't create

Parameter	Description
path	Defines the CSDB path to the file and domain to which the package will be published, for example, PRIMARY.SOFTWARE.
attr CLASS {body}	<p>Additional instance attribute values to be added during the publishing process. Note that the Publisher only supports PACKAGE and ZSERVICE as values for CLASS. Any attribute of these classes can be passed, even those not supported by the user interface. Examples:</p> <pre data-bbox="539 499 1096 782"> attr ZSERVICE { AUTHOR {John J. User} URL {http://www.nowhere.com} OWNER {Microsoft Corp} } attr PACKAGE { RELEASE {1.0} } </pre> <p>Enclose values in braces.</p>
msi {body}	<p>This section specifies MSI user interface settings. Allowable settings are:</p> <ul data-bbox="576 925 1248 1579" style="list-style-type: none"> • <code>aiplocation</code> This specifies where the AIP is to be created (if needed). • <code>publishmode</code> Advanced or Basic. • <code>execui</code> How to execute the MSI's user interface during AIP creation. Allowable values are Full, Basic, or None. • <code>deleteaip</code> Whether or not to delete the AIP after publishing. Allowable values are "y" or "n". • <code>aiparguments</code> Optional arguments for AIP creation. • <code>enableperuser</code> Whether to disable per user installation of this package (N) or any combination of the following: S=shortcuts, R=registry, E=private exe, D=private dll. Specifying "SD", for example, will turn on per user installation for shortcuts and private dlls. • <code>additionalfiles</code>

Parameter	Description
	<p>Whether additional files should be included. Allowable values are “y” or “n”.</p> <ul style="list-style-type: none"> connectaiptosvc <p>Whether the AIP should be included. Allowable values are “y” or “no.”</p> <p>An example msi section would resemble the following:</p> <pre>msi { aiplocation {c:\testaip} publishmode advanced execu none deleteaip n aiparguments "" enableperuser sre }</pre> <p>Note: Values are case-sensitive.</p>

Here is a sample configuration file.

```
pkgname      MYTESTPKG
package     CJMTEST
pkgdesc     {This is a test package}
path        PRIMARY.SOFTWARE
addtosvc    1
service     CJMTESTS
svcname     MYTESTSVC
svcdesc     {This is a test service}
attr PACKAGE {
    RELEASE 1.0
    BEHAVIOR
}
attr ZSERVICE {
    AUTHOR      me
    URL         {HTTP://WWW.NOWHERE.COM}
    OWNER       {Microsoft Corp.}
    PRICE       {5}
}
msi {
    aiplocation {g:\aip}
    publishmode advanced
    deleteaip  n
    aiparguments {}
    enableperuser n
}
```

Defining which File to Publish on the Command Line

In addition to defining a configuration file, you can define the file to be published on the command line. Use the `-file` parameter to define which file to use in the publishing session.

```
nvdtk.exe pubport.tkd -cfg "c:\program files\test.cfg" -  
file "c:\program files\testfile" -user USER -password  
PASSWORD
```

Publishing Multiple .msi Files (Batch Mode)

The Publisher allows you to publish multiple `.msi` files to the CSDB during one publishing session. This is achieved by calling the Publisher from the command line and designating a location using the `-batchmsi` parameter. The Publisher then scans this location and all subdirectories for `.msi` files as well as any configuration files or transform (`.mst`) files.

▶ Unlike publishing single `.msi` files using the command line, when publishing batches of `.msi` files with the Publisher, the command line parameter `-cfg` cannot be used to designate a configuration file. Instead, at least one configuration file must be included within the `.msi` file directory location.

Directories are scanned until a conforming directory is found. Each directory should contain the components of a Client Automation package that will be published to the CSDB, including at least one `.msi` file as well as, optionally one configuration file and any applicable transform files. After a conforming directory has been processed, scanning continues to look for additional conforming directories that exist under the starting directory.

The files contained in the designated location must follow a few simple rules:

- If more than one `.msi` file exists in a directory, an error will be recorded and the publishing session scan will continue.
- If more than one configuration file exists in a directory, an error will be recorded and the publishing session scan will continue.
- If desired, only one configuration file can be used for all `.msi` files found. When only one configuration file is discovered in the root directory location, the Publisher will apply that configuration to all `.msi` files it finds thereafter. To define a unique name to each package then, use symbolic substitution, as described in [Customizing Configuration File Values with Symbolic Substitution](#) on page 106.

- In order for the scan to be successful, at least one configuration file must be present at the root directory location or co-located with any MSI that is found.
- The scan recurses until an MSI file is found. At that point, the MSI file is processed, and the scan does not recurse any deeper in that directory tree. Scanning continues at the parent directory of the MSI file that is found.
- If the “additionalfiles” keyword was defined with a ‘y’ value in the configuration file, any additional files at the level of the MSI file and below are included with the AIP (if advanced mode) or the package (if basic mode).
- In batch mode, any transform files co-located with the MSI file are included in the package.

A sample command line to run the Publisher and designate the location to scan would look like the following:

```
nvdtk.exe pubport.tkd -batchmsi "c:\msifileslocation" -user
USER -password PASSWORD
```

This example will cause the Publisher to scan the directory `c:\msifileslocation` and all its subdirectories looking for `.msi` files as well as any configuration files or transforms.

When the publishing session is finished, verify the session by referring to the `pubport.log` file located in your log directory.

Using Multiple Configuration Files

Only one configuration file is required for all the `.msi` files that you publish as long as you are using symbolic substitution to assign unique package names, as discussed in [Customizing Configuration File Values with Symbolic Substitution](#) on page 106. However, if you would like to apply a separate configuration file to each `.msi`, then a separate configuration file must reside within the directory of the `.msi` to which it will be applied.

Any configuration file found within a directory will be applied to the `.msi` file found within that directory and any subsequent `.msi` files found in each subdirectory, unless another configuration file is found.

For instance, the `Acrobat Reader 6.msi` file, as displayed in [Figure 5](#) on page 105, will use the configuration file `global.cfg`, located in the root location directory, `\Adobe MSI Apps to Publish`, since no configuration file is included in the `\Acrobat Reader 6.0` directory.

Figure 5 Root location directory that contains configuration file

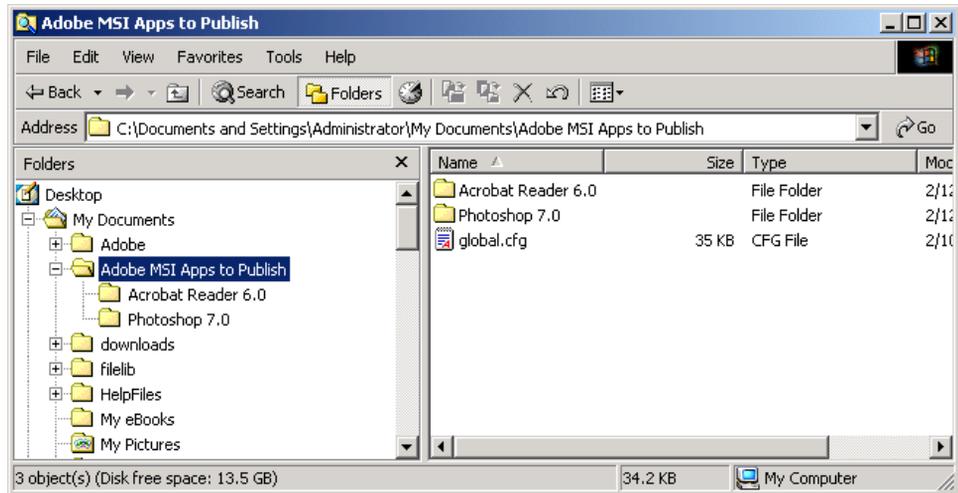
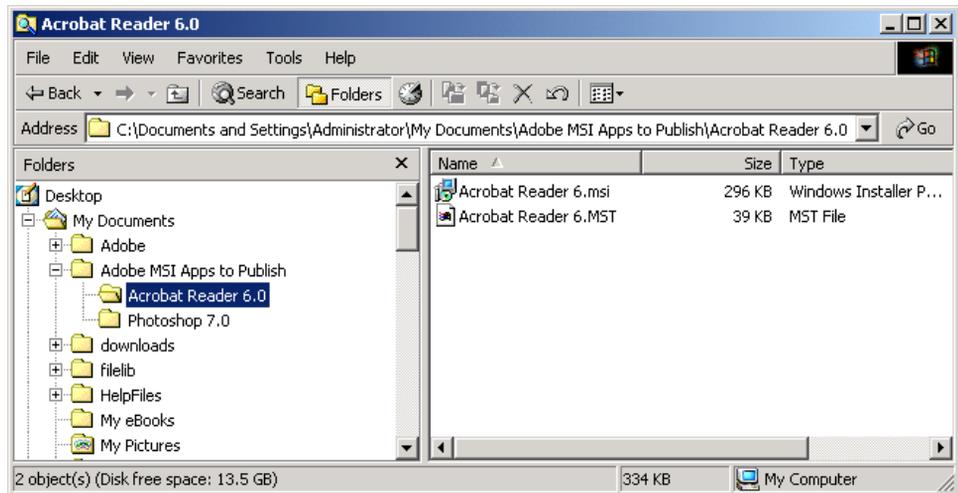


Figure 6 Configuration files within the directory structure



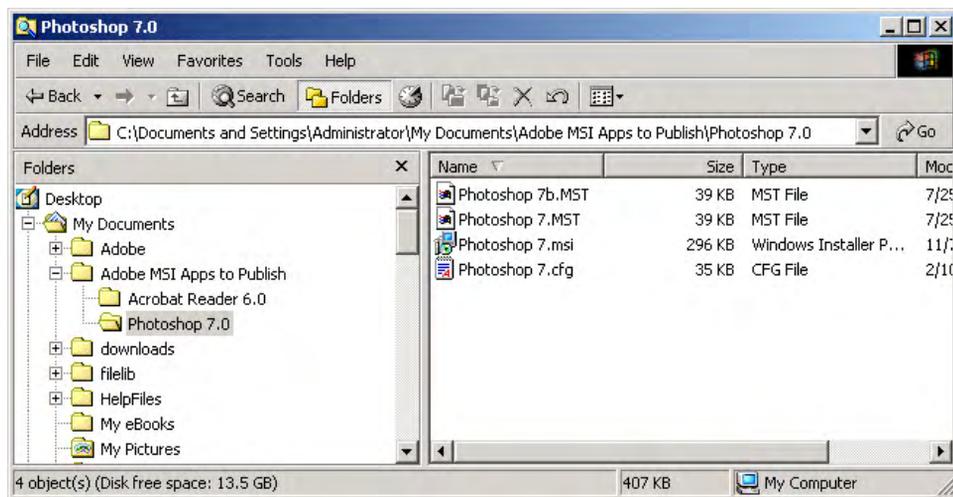
Including Transform (.mst) Files

You can apply any number of transform files with your .msi files by including those transforms in the same directory with the .msi file to which

they will be applied. Transform files are only applied to the .msi file with which they share a directory.

Figure 7 below displays two transform files, both of which will be included with the Photoshop 7.msi package upon publication.

Figure 7 Transform files included in the directory structure



Customizing Configuration File Values with Symbolic Substitution

If you want to publish similar .msi files and would like to use only one configuration file, you can direct the Publisher to use symbolic substitution to assign unique values. For instance, to create unique package names set the pkgname variable within the configuration file to %MSIPROP.VARNAME% or another symbolic variable as found in your .msi file. When used, the Publisher will search the current individual .msi file it is publishing and substitute the pkgname value with the corresponding value found within the .msi file, based on the symbolic value you supplied.

See the sample configuration file below that displays the use of symbolic substitution.

 You cannot use symbolic substitution for the path attribute.

```
# -----  
# Publisher - Sample imbedded MSI Properties Configuration File  
# -----  
#  
# Package related variables
```

```

#-----
pkgname          {%MSIPROP.RADIA_PKG_DISPLAY_NAME%}
package          {%MSIPROP.RADIA_PKG_INSTANCE_NAME%}
pkgdesc          {%MSIPROP.RADIA_PKG_DESCRIPTION%}
path             {PRIMARY.SOFTWARE}
#-----
#      Service related variables
#-----
addtosvc         {%MSIPROP.RADIA_ADDTOSVC%}
service          {%MSIPROP.RADIA_SVC_INSTANCE_NAME%}
svcname          {%MSIPROP.RADIA_SVC_DISPLAY_NAME%}
svcdesc          {%MSIPROP.RADIA_SVC_DESCRIPTION%}

attr PACKAGE {
    RELEASE      {%MSIPROP.RADIA_PKG_RELEASE%}
}

attr ZSERVICE {
    AUTHOR       {%MSIPROP.RADIA_SVC_AUTHOR%}
    URL          {%MSIPROP.RADIA_SVC_URL%}
    OWNER        {%MSIPROP.RADIA_SVC_OWNER%}
    PRICE        {%MSIPROP.RADIA_SVC_PRICE%}
    ZSVCMO       {%MSIPROP.RADIA_SVC_ZSVCMO%}
    ZCREATE      {%MSIPROP.RADIA_SVC_ZCREATE%}
}

msi {
    aiplocation  {%MSIPROP.RADIA_AIPLICATION%}
    publishmode  {%MSIPROP.RADIA_PUBLISHMODE%}
    execuicui    {%MSIPROP.RADIA_EXECUI%}
    deleteaip    {%MSIPROP.RADIA_DELETEAIP%}
    aiparguments {%MSIPROP.RADIA_AIPARGS%}
    enableperuser {%MSIPROP.RADIA_ENABLEPERUSER%}
    additionalfiles {%MSIPROP.RADIA_ADDITIONALFILES%}
}

```

Notice in the example above that every value for each package and service that is created will be generated from the associated .msi file property table.

There are two primary MSI substitution value sources.

- `%MSISUMM_summary_variable_name%`
extracted from the summary information associated with the MSI file.
- `%MSIPROP_property_name%`
extracted from the property table associated with the MSI file.

Embedded MSI Properties and Extensions for Windows Installer User-Defined Properties

When you use the Extensions for Windows Installer to create an `.msi` package, you can automatically add a set of MSI properties and their default values. To do this use the Packager for Windows Installer → Windows Installer Options dialog box that is shown at the end of a Unified Packager session. Select the Information tab to define User-defined properties. You can manually add these properties to any package whether created by Wise, InstallShield, or another vendor. Refer to the *Extensions for WI User Guide* for more information.

Publishing Hardware Configuration Elements

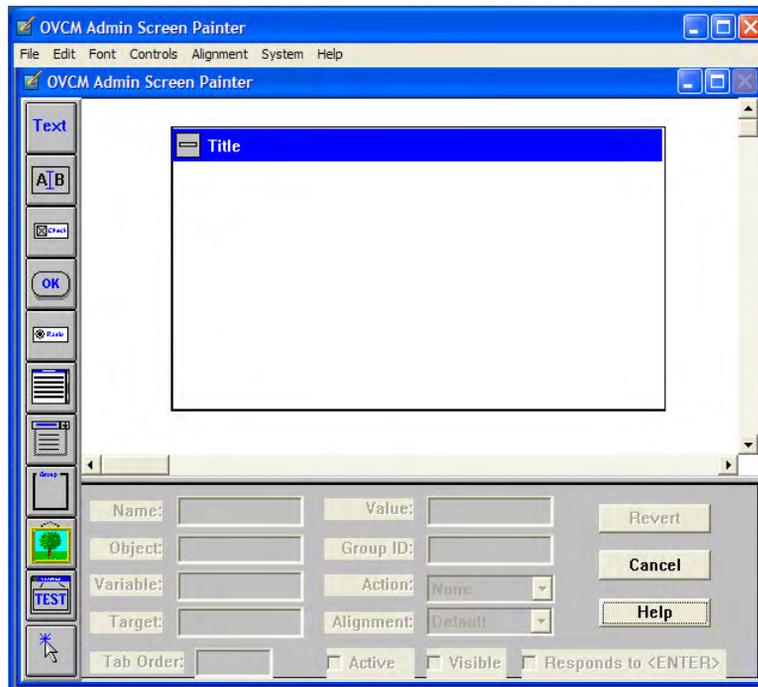
Use the Publisher to publish hardware configuration elements for distribution to devices in your environment. Refer to the *HP Client Automation OS Manager Hardware Configuration Guide (OS Manager Guide)* for detailed information. This guide contains detailed steps for publishing hardware configuration elements.

Screen Painter

This section contains a description of the Screen Painter, a tool that you use to design and create dialog boxes. It provides a graphical user-interface, with drag-and-drop capabilities, for creating and positioning typical controls, e.g., check boxes, list boxes, and buttons.

To open the Screen Painter

- From the **HP Client Automation Administrator** folder, click **Client Automation Administrator Screen Painter**. The Screen Painter window opens.



File Menu

Table 18 on page 110 describes the navigation features in the **File** drop-down menu.

Table 18 Screen Painter File menu

Feature	Function
New	Creates a new dialog box.
Open	Opens the Select a Panel Object File for Editing dialog box. From the list, select a panel object to edit.
Close	Closes the current editing session.
Save	Saves the current panel object with the same name.
Save As	Saves the current panel object with a new name.
Revert	Reverts to a previously saved version of the panel object.
Page Setup	Select a portion of the current dialog box to make this option available. It is the standard Windows Print Setup dialog box.
Print	Allows you to print a copy of the dialog box. Note: Screen Painter's Print function prints the current dialog box pixel-by-pixel, with one printer dot per pixel. If you have a high-resolution printer set at 600 DPI, the resulting printout will be illegibly small. To fix this, use the Properties button of the Page Setup dialog box to reduce the printer resolution to approximate screen resolution. This will produce a printout roughly the same size as that which appears on the screen.
Exit	Closes the Screen Painter.

Edit Menu

The Edit menu contains the standard Windows editing functions.

Font Menu

The Font menu enables you to change the font of the text displayed for one or more selected dialog box elements.

Controls

The following instructions explain how to add and set controls in a dialog box, and how to add or change the title of a dialog box. Use either the Controls toolbar or the Controls menu.

Controls Toolbox

The Controls toolbox provides buttons to facilitate the process of adding controls to a dialog box. The following is a table of the buttons and a brief description of the function of each control associated with it.

Table 19 Controls toolbox

Button	Button Name	Function
	Static Text	For displaying labels, descriptions, and object variable values.
	Edit	For entering text and displaying object variable values.
	Check Box	For Yes/No or On/Off types of controls.
	Push Button	For performing an action such as EXIT or displaying another dialog box.
	Radio Button	For a multiple choice option where only one item can be selected.
	List Box	For a scrollable list of choices where only one item can be selected.
	List Button	For a drop-down list of choices where only one item can be selected.
	Group Box	For grouping and organizing controls.
	Image	For displaying a bitmap image.
	Test	For executing the dialog box in real-time mode.

Button	Button Name	Function
	Select	For selecting and modifying dialog boxes, controls, and properties.

Properties Box

Use the Properties box to assign properties to each element in the screen object. You will not need to assign all these properties to every control in your screen object. Some controls need only a few properties. Use the Properties box at the bottom of the Screen Painter window to add properties.

Figure 8 Screen Painter Properties box

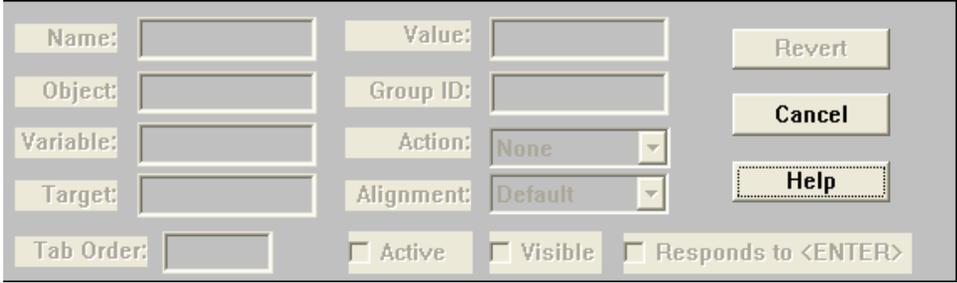


Table 20 below describes these properties.

Table 20 Properties box elements

Property	Description
Name	The label displayed on the control. Also used for indicating a target name.
Object	The object from which the control reads and writes data.
Variable	The variable from which the control reads and writes data.
Target	The dialog box, file, or control that is affected by what is specified in the Action field.
Value	Enabled only for Check Box, Radio Button, and List Box. See these sections for an explanation of specifications for this field.
Group ID	A user-specified identifier that enables you to group together several controls.

Property	Description
Action	Specifies what the control will do when selected.
Alignment	Justification for the text entered in the Static field of the Static Text control. Push Button text defaults to 'center'.
Tab Order	Specifies the order in which the control is reached when the user presses Tab . The default is the order in which the controls were added to the dialog box.
Active	Determines whether the control will be initially selectable or dimmed. (default: Active.)
Visible	Determines whether the control will be initially displayed or hidden. (default: Visible.)
Responds to <Enter>	Allows the user to choose the highlighted Push Button by pressing Enter . This feature works only with Push Buttons, and only one per dialog box.
Revert	Restores the property values for the control to when the control was last updated.
Cancel	Closes the editing session without saving changes.
Help	Invokes on-line help (not available yet).

Controls Menu

This section describes the controls that are available on the Controls menu.

Deleting a Control

This section describes how to delete controls.

To delete a control

- Select one or more controls and press **Delete** (on the keyboard).

Static Text

Use this control to label other controls, or add descriptions or instructions to the dialog box.

To insert static text

- 1 Select the Static Text tool from the toolbar or **Static Text** from the Controls menu.
- 2 Click-and-drag a rectangle in the dialog box. Re-size this control later, if needed.
- 3 Click on the static text control just created.
- 4 In the Properties Box, specify values for the active fields.

For the Static text control, the Name field becomes the Static field. [Table 21](#) below contains a description of it.

Table 21 **Static field**

Field	Description
Static	This field enables you to enter static text to label a control (or controls) in this dialog box. Leave blank if text is coming from a variable in an object.

Edit

Use this tool to add a text box. A text box is a control that enables the entry and display of variable data.

To insert a text box

- 1 Select the Edit tool from the toolbar, or **Edit** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click on the text box just created.
- 4 In the Properties Box, specify values for the active fields.

For the Edit control, the Name field becomes the Edit field. [Table 22](#) below contains a description of it.

Table 22 **Edit field**

Field	Description
Edit	The name of this control for referencing in groups.

Push Button

A push button enables the user to initiate an action.

To insert a push button

- 1 Select the Push Button tool from the toolbar or **Push Button** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click on the push button just created.
- 4 In the Properties Box, specify values for the active fields.

For the Push Button control, the Name field becomes the Push field.

[Table 23](#) below contains a description of it.

Table 23 Push field

Field	Description
Push	The text label displayed on the button.

The following table describes the actions that you can assign to a push button.

Table 24 Actions for Push Buttons

Action	Description
EXIT	Saves the contents of the dialog box, and closes it.
CANCEL	Closes the dialog box without saving changes.
SHOW	Toggles the control whose name is specified in the Target field to be visible or invisible. If the Target field contains a Group ID for a group of controls in the dialog box, then that group of controls is made invisible or visible.
ENABLE	Toggles the control whose name is specified in the Target field to be enabled (i.e., capable of becoming the focus of input) or disabled (i.e., dimmed, and incapable of becoming the focus of input). If the Target field contains a Group ID for a group of controls in the dialog box, then that group of controls is enabled or disabled.

Action	Description
PANEL	<p>Exits the current dialog box and displays the dialog box specified in the Target field. If the Target field is blank, then the Object and Variable fields specify a variable that contains the name of the dialog box to be displayed.</p> <p>You can use a push button in conjunction with a list box to enable the user to select which dialog box to display next from a list. The user selects an item from the list box, and presses the associated push button to launch his choice. To accomplish this, the push button's Target field must contain the name of a list box, and the push button's Object and Variable fields identify some variable in the multi-heap object that is the data source for the list box. In each heap of the list box's data source object, this variable contains the name of the panel object corresponding to the dialog box to display next, for the user choice associated with data displayed in the list box from that heap.</p>
LAUNCH	<p>Launches the application specified in the Target field. If the Target field is blank, then the Object and Variable fields specify the variable that contains the path and file name of the file to be launched.</p> <p>You can use a push button in conjunction with a list box to enable the user to select which program to launch from a list. The user selects an item from the list box, and presses the associated push button to launch his choice. To accomplish this, the push button's Target field must contain the name of a list box, and the push button's Object and Variable fields identify some variable in the multi-heap object that is the data source for the list box. In each heap of the list box's data source object, this variable contains the path and file name of the program to launch for the user choice associated with data displayed in the list box from that heap.</p>
PREVHP	<p>Pages to the previous heap in a multi-heap object (if not on the first heap) and reloads all the controls that use that object. The object to be paged is specified in the Object field. The variable specified in the Value field is displayed.</p>

Action	Description
NEXTHP	Pages to the next heap of a multi-heap object (if not on the last heap) and reloads all the controls that use that object. The object to be paged to is specified in the Object field. The variable specified in the Value field is displayed.
LNCHEX	Launch and Exit. Does the same thing as the LAUNCH action, and also exits the dialog box.
STDFILE	Presents the standard Windows Open File dialog box and places the user's selection (with full path information) into the variable specified by the Object and Variable fields.
STDFOLD	Presents the standard Windows Select Folder dialog box, and places the user's selection (with full path information) into the variable specified by the Object and Variable fields.

Check Box

A check box enables the user to indicate a binary (i.e., a yes/no) choice.

To insert a check box

- 1 Select the Check Box tool from toolbar or **Check Box** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. You can resize this control later.
- 3 Click on the check box just created.
- 4 In the Properties Box, specify values for the active fields.

For the Check Box control, the Name field becomes the Check field. [Table 25](#) below contains a description of it.

Table 25 Check field

Field	Description
Check	The label for the check box and the name of this control when referenced as a target by other controls.

Radio Button

A radio button enables the user to click one item from a set. Insert radio buttons individually, but group them in order to establish a set.

To insert a radio button

- 1 Select the Radio Button tool from the toolbar or **Radio Button** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click on the radio button just created.
- 4 In the Properties Box, specify values for the active fields.

For the Radio Button control, the Name field becomes the Radio field.

Table 25 below contains a description of it.



When created, a radio button defaults to a preset size. If the text entered exceeds the limits of the text box, expand it horizontally. The gray area will retain its size, but the full text will be displayed to the user.

Table 26 Radio field

Field	Description
Radio	The label for the radio button. Also, the name of this control when referenced as a target by other controls.

Group Box

A group box enables you to visually group a set of controls within a defined area of the dialog box.

To insert a group box

- 1 Select the Group Box tool from toolbar or **Group Box** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click on the group box just created.
- 4 In the Properties Box, specify values for the active fields.

For the Group Box control, the Name field becomes the Group field. [Table 27](#) below contains a description of it.



First create and align the controls, then place a group box around them. This will save time if you decide to reposition, add, or delete controls that will be grouped.

Table 27 Group field

Field	Description
Group	The text label for the group box.

List Box

A list box is a scrollable box from which the user may select only one item at a time.

To insert a list box

- 1 Select the List Box tool from toolbar or **List Box** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click on the list box just created.
- 4 In the Properties Box, specify values for the active fields.

For the List Box control, the Name field becomes the List Box field. [Table 28](#) below contains a description of it.

Table 28 List Box field

Field	Description
List Box	The name of the control for specifying it as a target, and the name of this control when referenced as a target by other controls.

List Button

A list button is a drop-down list control that enables the user to select one of the choices presented.

To insert a list button

- 1 Select the List Button tool from toolbar or **List Button** from the Controls menu.
- 2 In the dialog box, click-and-drag a rectangle. Re-size this control later, if needed.
- 3 Click the list button just created.
- 4 In the Properties Box, specify values for the active fields.

For the List control, the Name field becomes the List field. [Table 29](#) below contains a description of it.

- In order for your List Button to have drop-down capability, you must expand the control vertically in the design space.

Table 29 List field

Field	Description
List	The name of the control for specifying it as a target. Also, the name of this control when referenced as a target by other controls.

Image

To insert a bitmap image

- 1 Select the Bitmap Image tool from toolbar, or **Image** from the Controls menu.
- 2 Click-and-drag a rectangle proportionate to the image it will display. Re-size this control later, if needed.

- The image is scaled to fill the entire frame.

- 3 Click the frame just created.
- 4 In the Properties Box, specify values for the active fields.

For the Image control, the Name field becomes the Image field. [Table 30](#) on page 121 contains a description of it.

Table 30 Image field

Field	Description
Image	The full path of the bitmap image to be displayed, including the .bmp extension.

Select

The mouse-cursor functions as a select (arrow) tool. Click a control in the dialog box with the cursor in this mode, and Screen Painter highlights it with handles.

To resize a control

- Grab one of the control's handles and drag it to the desired size.

To reposition a control

- Click-and-drag the item (not by the handles) to the desired location.

Set Title

Name a dialog box to make it easier for the user to remember its function.

To give the dialog box a title

- 1 From the Controls menu, click **Set Title**, or double-click the blue title bar of the dialog box.
- 2 Type the title in the dialog box, and press **Enter**.

Alignment Menu

Arrange the controls you have inserted for a more legible appearance. The following table explains the nine options available for aligning controls.

The following instructions explain how to align controls. Users will find the dialog boxes easier to use if the controls are arranged in an orderly fashion. These features are available only from the Alignment menu.

Table 31 Alignment Functions

Feature	Function
Align Tops	Aligns the selected controls to the top edge of the uppermost selected control.
Align Bottoms	Aligns the selected controls to the bottom edge of the lowest selected control.
Align Lefts	Aligns the selected controls to the left edge of the leftmost selected control.
Align Rights	Aligns the selected controls to the right edge of the rightmost selected control.
Align Centers	Centers all selected controls on an imaginary vertical line that bisects the right edge of the right-most selected control and the left edge of the left-most selected control.
Align Same Height	Vertically resizes all selected controls to the same height as the (first) selected reference control.
Align Same Width	Horizontally resizes all selected controls to the same width as the (first) selected reference control.
Align Even Vertical	Adjusts the spacing among a group of selected controls so that there is an identical amount of vertical space between each pair of adjacent controls.
Align Even Horizontal	Adjusts the spacing among a group of selected controls so that there is an identical amount of horizontal space between each pair of adjacent controls.

 At least three controls must be selected for the Align Even Vertical and Align Even Horizontal functions to have any effect.

To align controls

The first five options of the Alignment menu enable you to reposition controls for a standard appearance.

- 1 Drag-and-drop a reference control to the desired position.
- 2 Select at least three controls that you want to align with the reference control by holding down **Shift** and clicking each one.

- To align controls vertically, from the Alignment menu, click **Align Lefts** or **Align Rights**.
- To align controls horizontally, from the Alignment menu, click **Align Tops** or **Align Bottoms**.
- To align controls by their centers, from the Alignment menu, click **Align Centers**.



Align controls prior to placing a Group Box around them. Any aligning or positioning done now can be revised at a later time.

To adjust a control's height and width

You can resize the controls for a consistent appearance.

- 1 Select a control to be used as a reference. (This control's size will not change. The other controls will resize to the same height or width as this reference control.)
- 2 To select the remaining controls you want to resize, hold down **Shift** and click each one.
- 3 From the Alignment menu, click **Align Same Height** or **Align Same Width**.

To space controls

You can set the spacing between controls for a consistent appearance.

- 1 Hold down **Shift** and click at least three controls.
- 2 From the Alignment menu, click **Align Even Vertical**, or **Align Even Horizontal**.

System Menu

The System menu contains choices for testing the dialog box.

Periodically, during the dialog box creation, and especially after it is finished, it is a good idea to test it to ensure that it works as you designed and intended. Doing a test runs the Panel Manager program, `edmpnlwr.exe`, using the current dialog box. This shows how the dialog box will look when it is displayed in production.

There are two methods, which are discussed below, for testing a dialog box:

- Common

- REXX Processing

Common

To test a screen using the common method

- 1 Click the Test tool from the Controls toolbar or **Test** from the System menu.
- 2 To end the test, click the displayed dialog box's close icon, or click a push button whose action terminates the dialog box.

REXX Programming

To test a screen using the REXX processing

- 1 Click **Use REXX for Test** from the System menu.
- 2 To end the test, click the displayed dialog box's close icon, or click a push button whose action terminates the dialog box.

Do Not Resolve Panel

To display variable names without resolving objects

- 1 Specify the symbolic notation in the Name field of the controls that you want left unresolved. Note that Client Automation always resolves any data specified in the Object and Variable fields in the Properties Box.
- 2 From the System menu, click **Do Not Resolve Panel**.

Help Menu

About is the only function that currently works. It displays the Client Automation version number.

CSDB Editor

This section describes how to use the CSDB Editor to manage the CSDB.

Getting Started

This section describes what you need to know and do before you start to use the CSDB Editor.

To open the CSDB Editor

1 Click **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator CSDB Editor**. The logon dialog box opens.

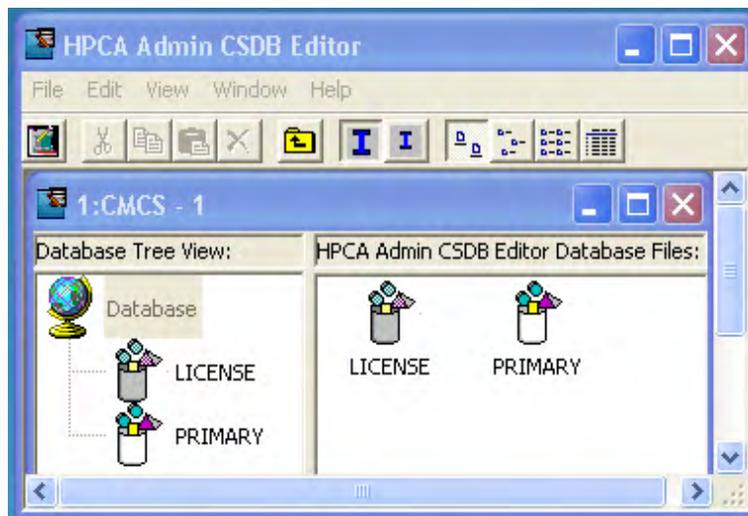
2 Type your User ID and Password.



The factory set user ID is RAD_MAST. No password is necessary. This might have been changed during installation.

If your administrator has configured the CSDB Editor to require passwords, after a successful logon using the assigned password, you can change a password by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

3 Click **OK**. The initial CSDB Editor window opens.



The CSDB Editor Window

This is the main window for the CSDB Editor. This section describes how to customize how it looks, and how to use each of its menus.

Customizing the CSDB Editor Window

You can customize the look of the CSDB Editor and how it functions by using options from the shortcut menus. The most frequently used functions are available from the CSDB Editor Toolbar. See [The CSDB Editor Toolbar](#) on page 128 for additional information about these functions.

CSDB Editor Window Menus

This section describes each of the menus in the main CSDB Editor window.

The File Menu

Use these options to open additional CSDB Editor windows or to close the current one. Multiple windows can be open at the same time. You can drag items between multiple windows.

The Edit Menu

Use these options to edit the CSDB. The first six choices apply to the selected items in the active pane or dialog box.

Table 32 Edit Menu options

Operation	Description
New	Creates a new occurrence of the selected item such as domain, class, or instance.
Cut	Removes the selected text and places a copy of it on the Windows clipboard.
Copy	Copies the selected text to the Windows clipboard.
Paste	Copies the current content of the Windows clipboard to the current cursor location.
Rename	Changes the friendly name and/or internal name of the selected item.
Delete	Removes the selected items from the CSDB.
Select All	Selects all of the instances in the list view of a CSDB Editor window.

Operation	Description
Invert Selection	Toggles the selection status of the instances in the list view. Those that are selected are cleared and those that are cleared are selected.
Find	Searches the Name and Instance Name columns in the selected class for instances that contain the text you specify. The results of the search appear highlighted and sorted at the top of the list in the list view.

The View Menu

These options control the appearance of the CSDB Editor. Show or hide the toolbar and status bar, control the size and display of the icons, refresh the window, or set the display and operational options from within this menu.

Table 33 View Menu options

Option	Description
Toolbar	Toggles the toolbar display on and off.
Status Bar	Toggles the Status Bar display on and off.
Tree Icons	Toggles the display between large and small icons.
List View	Choose from among four different displays for your list view—Large Icons, Small Icons, List, and Details.
Line Up Icons	Redraws the list view arranging the icons either vertically or horizontally.
Refresh	Redraws the list view of the selected CSDB Editor window, lining up all of the elements for easy reading.
Options	Opens the CSDB Editor Options dialog box. See Setting Preferences: The Options Dialog Box on page 129 for more information.
Font	Opens a dialog box that allows you to change the font used to display text in the CSDB Editor windows.

The Window Menu

Use these options when you have more than one CSDB Editor window open.

Table 34 Window Menu options

Option	Description
Cascade	Organizes and overlaps the windows starting at the top, left corner of the CSDB Editor interface.
Tile Horizontal	Automatically sizes open windows to the width of the CSDB Editor user interface. It allocates the available height within the CSDB Editor interface equally, stacking the windows one on top of the other.
Tile Vertical	Automatically sizes open windows to the height of the CSDB Editor interface. It allocates the available width equally, stacking the windows next to each other.
Arrange Icons	Docks minimized CSDB Editor windows in the bottom left corner of the interface.
1 1:CSname – 1	Multiple View allows the administrator to toggle between open CSDB Editor sessions. A check mark next to a session indicates which window is active.

The Help Menu

Get details about the CSDB Editor or link to the HP Software web site.

Table 35 Help Menu options

Option	Description
Contents	Opens a traditional Windows-style help interface.
Hewlett-Packard on the Web	Opens a sub-menu with links to the Hewlett-Packard web site pages for Technical Support and Home Page. Selecting one of these links opens your default Web browser and connects to the selected page over the Internet.
About CS Database Editor	Displays information about the current CSDB Editor—version number, Configuration Server information, and so forth. Click System Info to display a standard Microsoft System Information window. This application analyzes your machine and shows the results in an easy to navigate format.

The CSDB Editor Toolbar

The CSDB Editor toolbar provides easy access to frequently used functions:

Figure 9 CSDB Editor toolbar



Table 36 below describes the functions of these toolbar buttons.

Table 36 CSDB Editor toolbar buttons

Button	Function
	Open a new CSDB Editor window.
	Not available.
	Copy the selected item.
	Not available.
	Delete the selected item.
	Display the database level above.
	Display large icons in the tree view.
	Display small icons in the tree view. Maximizes the number of items that can be seen in the list view of the CSDB Editor at one time.
	Display large icons in the list view.
	Display small icons in the list view. Maximizes the number of items that can be seen in the list view of the CSDB Editor window at one time.
	Display the list view vertically. Maximizes the number of items that can be seen in the list view of the CSDB Editor window at one time.
	Display the list view vertically. Adds a column for TYPE (such as Class and Instance). Maximizes the number of items and provides detailed description of the list view.

Setting Preferences: The Options Dialog Box

The CSDB Editor has an Options dialog box that allows you to control some of the features of the CSDB Editor.

- To open the Options dialog box, click **View** → **Options**.

There are five tabs on the Options dialog box:

- General
- Instance Options
- Advanced
- KB AutoImport
- Notify

- Click a tab to access its configuration options.

General Tab

Figure 10 CSDB Editor Options General tab

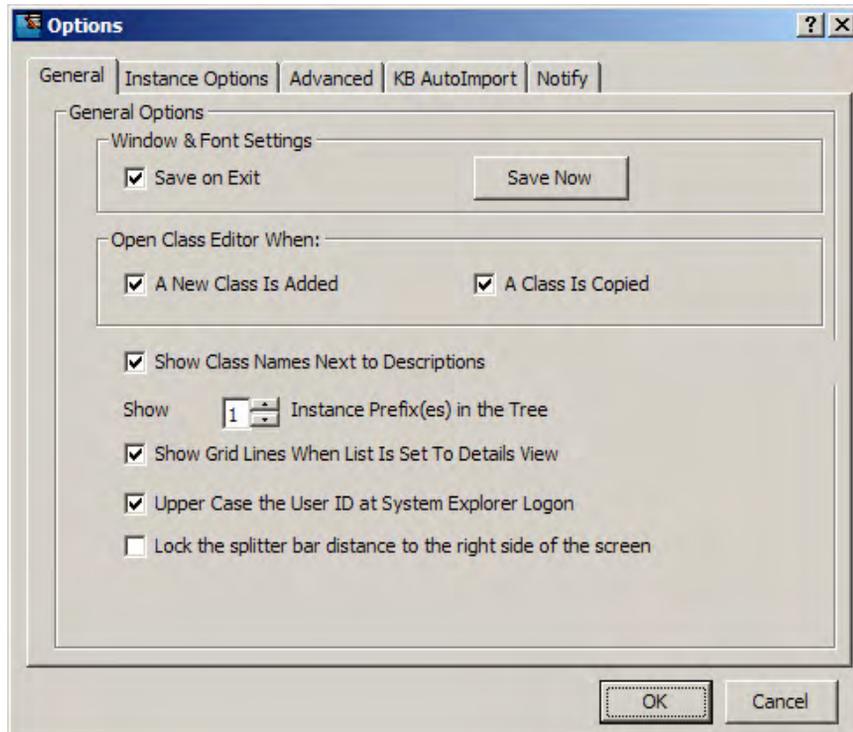


Table 37 describes the General tab options.

Table 37 Functions of the Options General tab

Option	Function and Use
Window & Font Settings	<p>Save on Exit tracks certain window settings and the font to use for displaying text when you exit from the CSDB Editor session.</p> <p>The window settings include the view setting (i.e., large icons, small icons, list, or details) for the tree and list views of the CSDB Editor window, and the relative width of each pane. Window settings are saved independently for each level of the CSDB.</p> <p>Save Now saves the current window settings and font in use.</p> <p>To change the font, choose a font from the dialog box that opens when you click View → Fonts from the menus.</p>
Open Class Editor When	<p>A New Class is Added opens the Editing Class dialog box when you add a class.</p> <p>A Class is Copied opens the Editing Class dialog box when you copy a class.</p> <p>Since you will probably want to edit the newly copied or newly added class, HP recommends that you have these two check boxes selected.</p>
Show Class Names Next to Descriptions	<p>Shows the Client Automation internal name for classes next to their friendly names in the tree view and in the list view. The internal name will appear in parentheses next to the friendly name.</p> <p>This is useful if you usually do not use the Details View for the list view. The Details View in the list view window always shows the internal name for classes in the Type column.</p>
Show <i>n</i> Instance Prefix(es) in the Tree	<p>This option can be set to 0, 1, 2, or 3 to control how many levels of instance name prefixes (delimited by the underscore character) will be compressible and expandable in the tree view. This is described in Navigating the CSDB Editor on page 137.</p>
Show Grid Lines When List Is Set To Details View	<p>Displays grid lines in the list view when its view option is set to Details.</p>
Uppercase the User ID at CSDB Editor Logon	<p>Automatically converts the user ID into all uppercase.</p>

Option	Function and Use
Lock the splitter bar distance to the right side of the screen	Locks the width of the right side of the CSDB Editor window during resizing. When this check box is not selected and the window is resized, the width of the left side of the window remains fixed, and the width of the right side of the window varies.

Instance Options Tab

Figure 11 CSDB Editor Options Instance Options tab

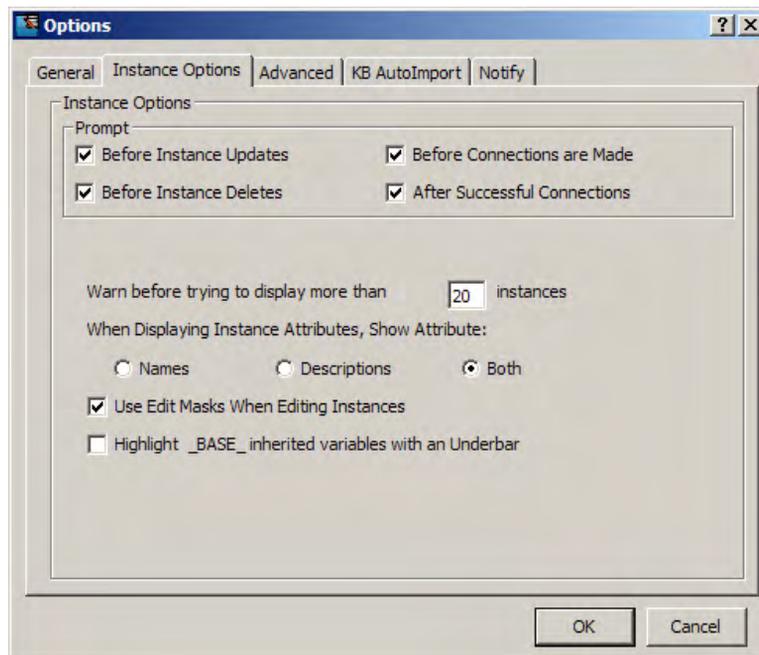


Table 38 describes the **Instance Options** tab options.

Table 38 Functions of the Options Instance Options tab

Option	Function and Use
Prompt	These four check boxes control whether CSDB Editor prompts you before completing the actions indicated by their descriptions.

Option	Function and Use
Warn before trying to display more than <i>n</i> instances	<p>In large desired states, certain classes such as FILE or USER can contain many instances. If you attempt to view the instances of a class that contains more instances than this option setting, a warning opens. This gives you the opportunity to filter the class instances, before they are displayed. This is a performance issue. If you try to display too many instances, you will wait while the CSDB Editor retrieves them from the CSDB, and you will have to scroll through a large list to find the instances you want to work with.</p> <ul style="list-style-type: none"> • Click Yes to open the Filter Class dialog box, and enter an appropriate filter. • Click No to display all of the instances. • Click Cancel to close the warning without displaying any instances.
When Displaying Instance Attributes, Show Attribute	<p>This setting determines which columns to display (Name, Description, or both) in the Details view on the list view of the window, when you open an instance by double-clicking its name in the tree view.</p>
Use Edit Masks When Editing Instances	<p>This setting determines whether the Edit Instance dialog box will be sensitive to the attribute type in the data entry area of the dialog box. If you select this check box, attributes that hold logical (i.e., Yes/No) data have check boxes in the data entry area. Attributes that have multiple choices have drop-down lists in the data entry area. A flag set will appear as a series of check boxes in the data entry area. You can set the mask by entering the valid choices for an attribute within square brackets at the end of the attribute's description in the class definition.</p> <p>If you do not select this check box, all data entry in the Edit Instance dialog box will be plain text.</p> <p>This facility is described in detail in Optional Attribute Editing Controls on page 152.</p>
Highlight <u>_BASE_</u> inherited variables with an Underscore	<p>Use this setting if you want to see which values in an instance are inherited from the <u>_BASE_INSTANCE_</u> of the class. The inherited instances will show with a red underscore.</p>



Figure 12 CSDB Editor Options Advanced tab

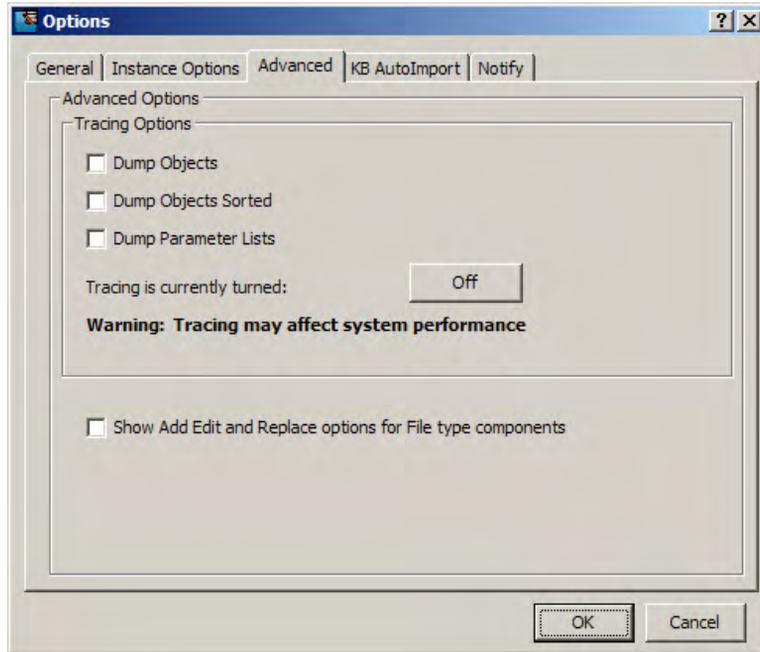


Table 39 describes the Advanced tab options.

Table 39 Functions of the Options Advanced tab

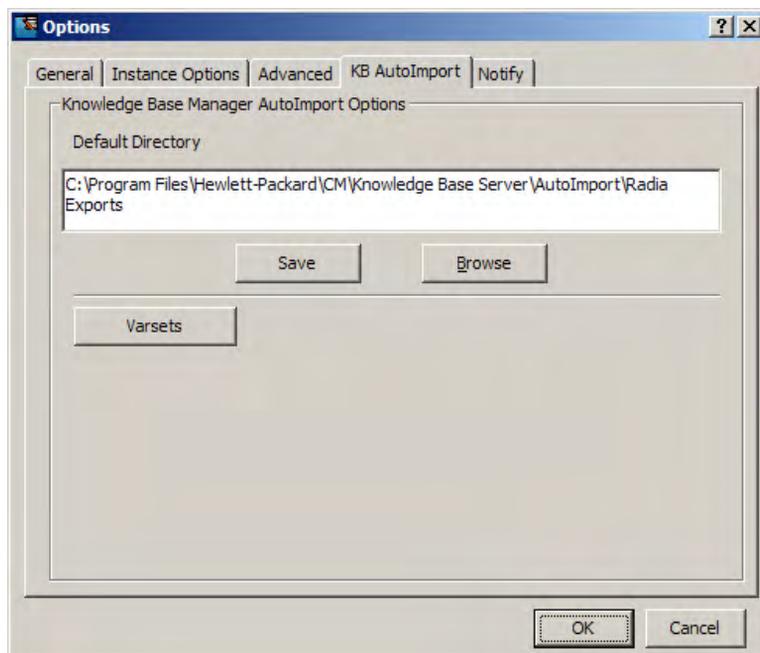
Option	Function and Use
Tracing Options	Use these settings only for troubleshooting the CSDB Editor.  Do not change these settings unless instructed to do so by HP Technical Support.
Show Add, Edit and Replace options for File type components	 Use this setting only in non-production environments. This setting determines whether to display the following commands on the shortcut menus for file-type components: <ul style="list-style-type: none"> • Edit this Component • Replace Component Data • Add a Component

For details, see Appendix B, [Naming Conventions](#).

KB AutoImport Tab

Use the KB AutoImport tab to define the default auto-import directory for the HP Client Automation Knowledge Base Server (KB Server). For details, refer to the *HP Client Automation Knowledge Base Server Installation and Configuration Guide (KB Server Guide)*.

Figure 13 CSDB Editor Options KB AutoImport tab



To specify the KB AutoImport options

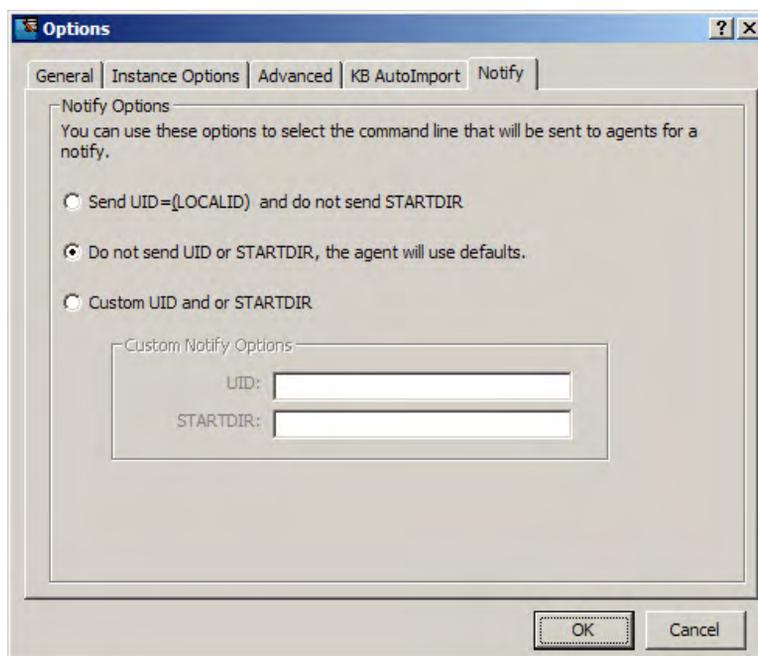
- 1 Specify or browse to the default directory to use to import **state files** into the KB Server database. The default location is an export directory installed by the KB Server. This directory includes the required variable set subdirectory, `\VarSets`.
 - ▶ Any directory that will be used as an export directory must include a subdirectory named `\VarSets`.
- 2 Click **Save** on the KB AutoImport tab. This ensures that the default directory that was specified for the KB Server default auto-import directory is set.

Notify Tab

Use the Notify tab to specify the `radskman.exe` command line values for UID and STARTDIR that will be sent to agents when they are notified. (The UID is used to identify the current session, and STARTDIR specifies the IDMLIB starting directory.) For example, select the **Custom UID and or STARTDIR** option to accept or enter values to notify agents by machine name, as opposed to the name of the user who is currently logged on.

The Notify options set here are used by the Notify Manager and are displayed on the Notify Options dialog box as shown in [Figure 14](#) below. For more information, see [Notify](#) on page 192.

Figure 14 CSDB Editor Options Notify tab



[Table 40](#) below describes the Notify tab options.

Table 40 Functions of the Options Notify tab

Option	Function and Use
Send UID=[LOCALID] and do not send STARTDIR	Creates a Notify command with UID=[LOCALID]. This command notifies agents by the name of the user who is currently logged on. To notify agents by the name of the machine, use the Custom UID and or STARTDIR option.

Option	Function and Use
Do not send UID or STARDIR, the agent will use defaults.	Creates a Notify command without any values for UID or STARTDIR; agents use the default values for UID and STARTDIR. To specify non-default values, use the Custom UID and or STARTDIR option.
Custom UID and or STARTDIR	Creates a Notify command with the UID and STARTDIR values entered in the Custom Notify Options text boxes. Use this option to notify agents by machine name, or to specify the appropriate starting directory. <ul style="list-style-type: none"> • Accept the Custom UID value of \$MACHINE to notify the agents by machine name. You can also enter another UID value to identify the current session using a custom value. • Accept the Custom STARTDIR value of SYSTEM to notify and install an application in the machine context. You can also enter another STARTDIR value.

Navigating the CSDB Editor

The CSDB is hierarchically organized and subdivided into files, domains, classes, and instances. The tree view enables you to view successively lower levels in the hierarchy, and easily locate and work with specific CSDB components.

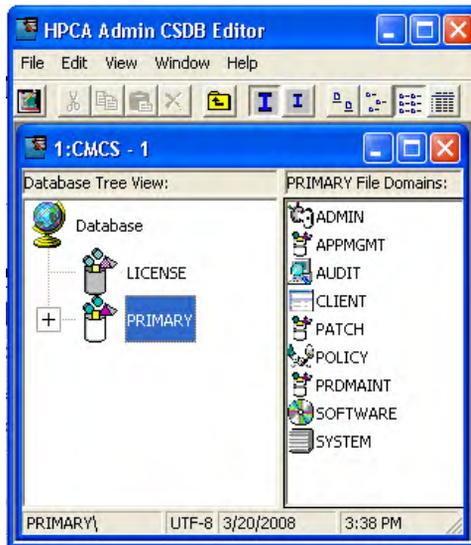
To navigate the tree structure

- 1 Open the CSDB Editor, and double-click **PRIMARY** in the tree view.



The PRIMARY File domains will vary according to which Client Automation products have been installed in your environment.

The CSDB Editor window will be similar to the following:



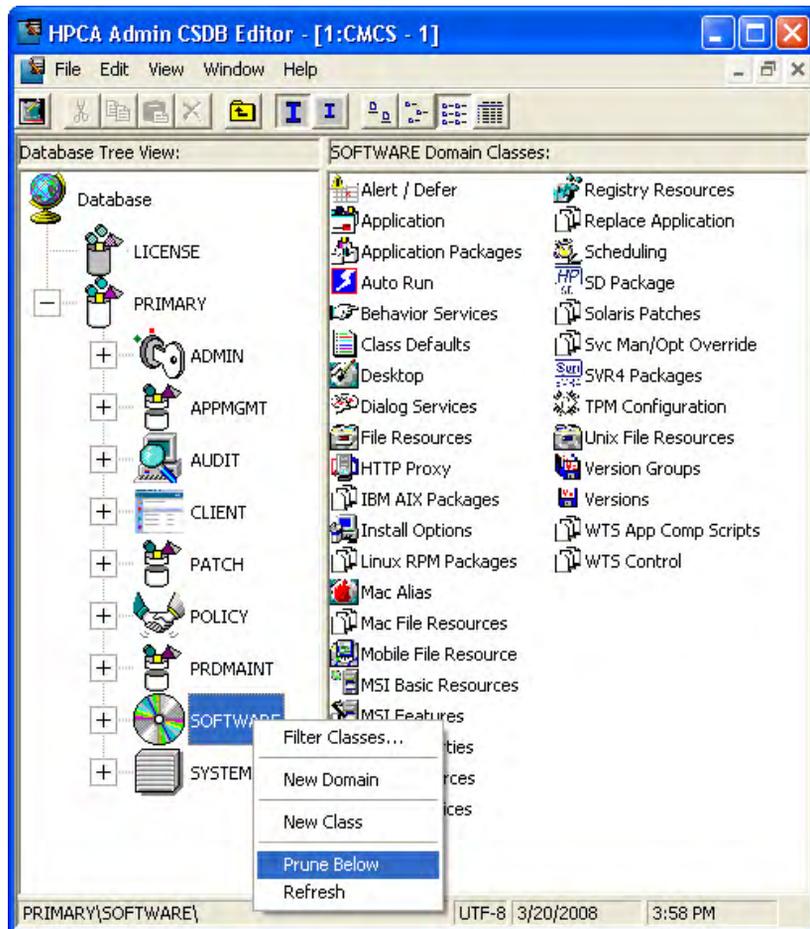
- 2 To open a CSDB component, double-click its icon.

Navigation History

Most navigation features in the CSDB Editor are similar to those in Windows Explorer. However, to discard the history of which branches were previously opened beneath a particular component, use the Prune Below function.

To use the Prune Below function

- 1 Right-click the SOFTWARE Domain icon in the tree view. A shortcut menu opens.



- 2 Click **Prune Below**. The tree view collapses all open nodes below the SOFTWARE Domain, and discards navigation history.

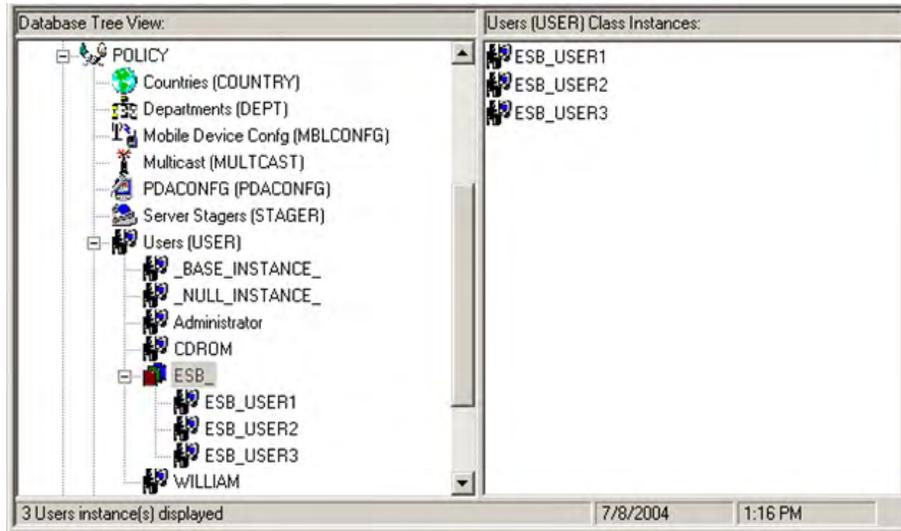
Compound Names

Use compound names to make navigation in the tree view more manageable. Client Automation provides one, two, or three collapsible levels, based upon one, two, or three high-level qualifiers of the instance name.

A compound name has one, two, or three prefixes, each separated from the rest of the name by an underscore character. For example, you might want to group your users according to the building where they are located. Instance names for these users will have a building identifier prefix. If you had users in the Empire State Building, you might assign ESB_ as the instance name prefix to identify them. Each user in the Empire State Building will be

assigned an instance name beginning with ESB_. The tree view will automatically add a level.

Figure 15 Compound names



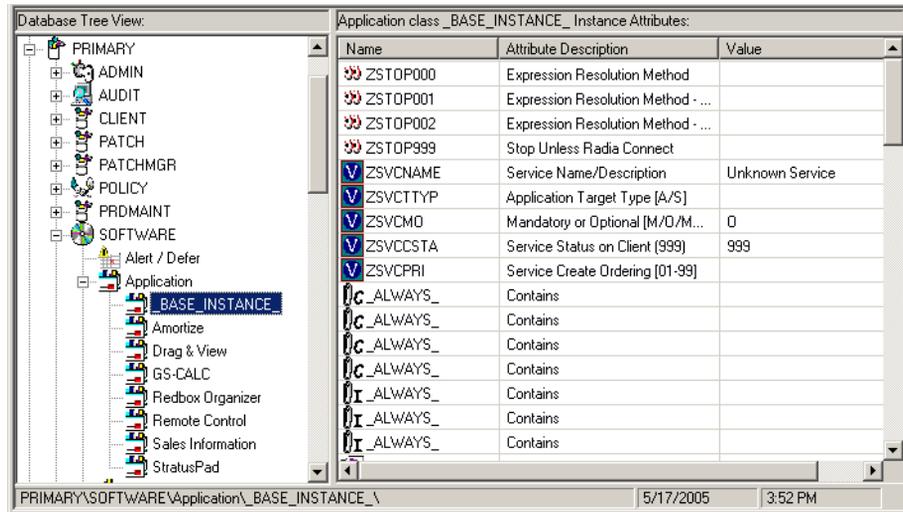
The number of prefix levels in an instance name (0, 1, 2, 3) that are displayed in the tree view is controlled by the Show n Instance Prefixes in the Tree option setting. This is found on the General tab of the Options dialog box and can be accessed from the View menu. The Options dialog box is described in [Setting Preferences: The Options Dialog Box](#) starting on page 129.

Each underscore in an instance name delineates a level for the tree view. Thus, the instance name ESB_FLOOR10_USER1 has two levels of prefix, which, if the setting Show n Instance Prefixes in the Tree were set to 2 or 3, would be displayed in two collapsible non-elementary levels (FLOOR10, within ESB).

Inheritance in the CSDB

Each class in the CSDB has a *base instance*, named `_BASE_INSTANCE_`. It provides the default values for the other instances of the class. Any attribute in an instance will inherit its value from the same attribute in the `_BASE_INSTANCE_` unless it has been specifically assigned another value.

Figure 16 BASE_INSTANCE_ of Application (ZSERVICE) Class



Inherited values appear in the instance the same as those values that you enter directly into the instance. You can tell if a variable's value is inherited from the BASE_INSTANCE_ by inspecting the BASE_INSTANCE_. Also, if you delete a value from a variable (see [Working with Classes](#) on page 142 to learn how), and the value re-appears in the instance automatically, the value is being inherited from the BASE_INSTANCE_.

Managing Domains, Classes, and Instances

In the CSDB Editor you can add, copy, edit, rename, and delete CSDB components (domains, classes, and instances). You can also connect instances.

You can access these functions from shortcut menus that open when you right-click targets in the CSDB Editor window. The menu that opens contains choices depending upon which target you right-click.

Working with Domains

This section describes how to add a domain to a file. Domain names can be a maximum of eight characters, and can include letters and numbers.

To add a domain to a file

- 1 Right-click the PRIMARY File or any domain, and select **New Domain** from the shortcut menu that opens. The Create Domain dialog box opens.
- 2 Enter a name for the new domain and click **OK**.

Working with Classes

This section describes how to add, edit, and delete a class.

Adding a Class

New classes can be added by

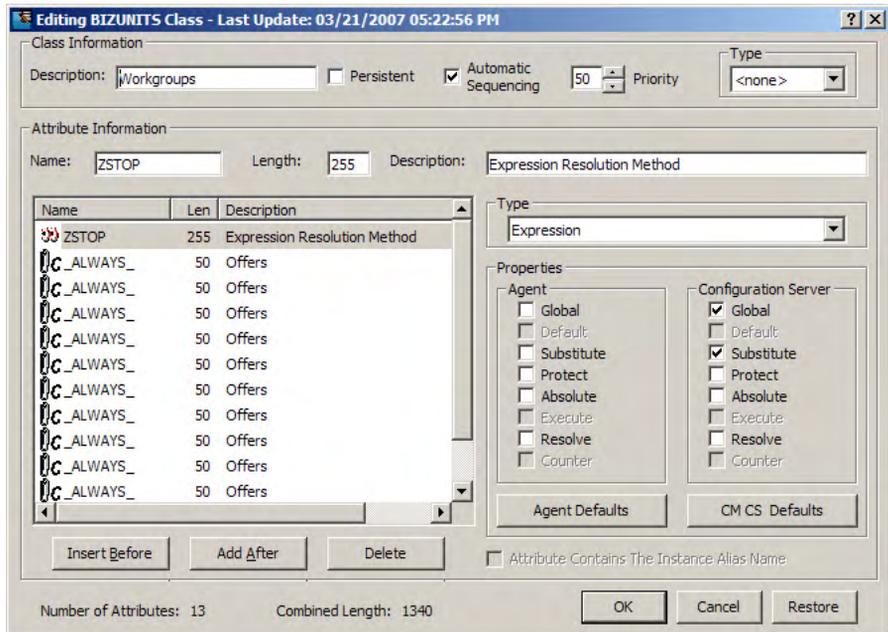
- copying an existing class,
or
- creating a new class.

The method you choose depends upon whether a class already exists whose attributes are substantially similar to the class that you want to add. If so, copy the existing class; otherwise, create a new one.

Copying a class creates a new class with the same attributes as the copied class, and a `_BASE_INSTANCE_` for the new class with its attribute values set as in the `_BASE_INSTANCE_` of the copied class.

To copy an existing class

- 1 Highlight the class to be copied and right-click it.
- 2 Select **Copy Class** from the shortcut menu. The Copy class dialog box opens.
- 3 Enter a name for the new class, and click **OK**.
 - If, when you set preferences in the Options dialog box (see [Setting Preferences: The Options Dialog Box](#) on page 129 for details), you did *not* choose to open the Class Editor when a class is copied, the original class is copied with the new name, the Class Editor is not opened, and no warning is issued. You will have to manually open the Class Editor to edit the new class.
 - Conversely, if you *did* choose to open the Class Editor, the Editing Class dialog box opens. See [Editing a Class](#) on page 145.



- 4 To complete the addition of the copied class, change the Description field to identify the copied class, edit the class to reflect the new class' requirements, and click **OK**.

The Class Edit Confirmation warning message warns you to back up your database before proceeding.

- 5 Click **Yes**. The new class addition is complete.



If you click **No** or **Cancel** in the Class Edit Confirmation warning message, the Class Editing dialog box closes, but the copying of the class does not terminate.

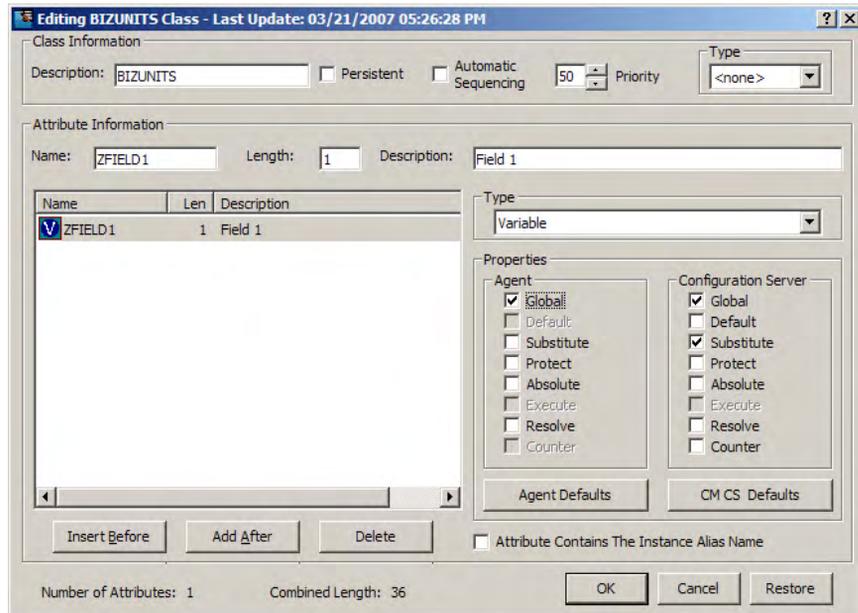
If you have created a class and want to delete it, see [Deleting a Class](#) on page 155.

After you create a class, check the MASTER_CONNECTION_RULES variable to determine if you need to establish a drag-and-drop rule for the new class. For additional information regarding connection rules and drag-and-drop connections, see [Maintaining Connection Rules in the CSDB](#) on page 202.

To create a new class

- 1 Highlight the domain in which you want to create the new class.
- 2 Right-click the domain name.

- 3 Select **New Class** from the shortcut menu. The Create Class dialog box opens.
- 4 Enter a name for the new class (BIZUNITS in this example), and click **OK**. The Editing Class dialog box opens. See [Editing a Class](#) on page 145.



- 5 To complete the addition of the new class, change the Description field to identify the new class, edit the class to reflect the new class' requirements, and click **OK**.
- 6 Click **Yes**. The new class is added.



If you click **No** or **Cancel** in the Class Edit Confirmation warning message, the Class Editing dialog box closes, but the copying of the class does not terminate.

If you created a class and want to delete it, see [Deleting a Class](#) on page 155.

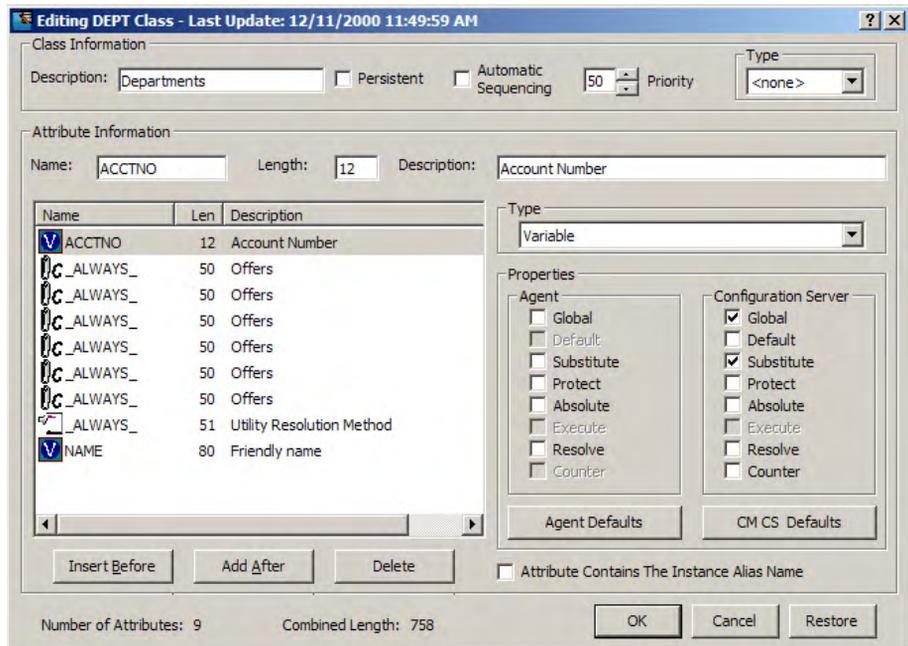
After you create a class, check the MASTER_CONNECTION_RULES variable to determine if you need to establish a drag-and-drop rule for the new class. For additional information regarding connection rules and drag-and-drop connections, see [Maintaining Connection Rules in the CSDB](#) on page 202.

Editing a Class

When you choose **Edit Class** from a shortcut menu, or during the addition of a new class as described in [Adding a Class](#) on page 142, the Editing Class dialog box opens.

To edit a class

- 1 Right-click the **Departments** Class in the tree view of the CSDB, and choose **Edit Class** from the shortcut menu. The Editing Class dialog box opens.



- 2 Set the Class Information properties at the top of the dialog box.

Description

The friendly name that describes the class and its purpose. This name appears next to the class icon in the tree view.

Persistent

This property should be used only by administrators who have a thorough understanding of the Client Automation resolution process.

A persistent class is one whose attributes and objects remain in global memory for the duration of the resolution process. The instances created

from persistent classes are available to the Client Automation agent for storage on its desktop.

If a class is not persistent, it is transient. A transient object exists only while it is being processed during resolution. Instances created from transient classes are not available to the Client Automation agent for storage on the desktop.

For more information about persistent and transient objects and how they are handled in the resolution process, see the *Essentials Guide* for a discussion of resolution.

Automatic Sequencing

Normally, resolution proceeds from attribute to attribute in the order the attributes appear in the Attribute List. See the *Essentials Guide* for a discussion of resolution.

Select this check box to impose a grouping on the order in which attributes of the class are processed during resolution:

- Expressions
- Variables
- Classes (connections)
- Methods

Priority

This field is two characters long and has a value between 00 and 99. During object resolution, the Configuration Server puts this value in the ZOPJPRI variable. If there is a blank or zero value in this field, the Configuration Server supplies the default value specified for that particular class. The default values for the classes are:

PATH	5
REGISTRY	60
DESKTOP	70
FILE	50
METACLAS	10
ZSERVICE	50
MACALIAS	70
All others	50

Type

Classifies classes according to their purpose and use. Client Automation uses this classification to affect the resolution process. For example,

resolution to populate the Service List on an agent computer does not process classes classified as Component classes.

Select one of the following values from the drop-down list:

- **<none>** – This class is not classified.
 - **Configuration** – This class identifies content to be managed by Client Automation on agent computers on an aggregate level. Typically, instances of this class do not have data that can be distributed. They are connected to instances of one or more Component classes. Examples: ZSERVICE, PACKAGE, VGROUP, VERSION, and so forth.
 - **Component** – This class identifies the items (files, registry entries, links, icons, etc.) that comprise the content identified by a Configuration class instance. Typically, instances of this class have data that can be distributed. Examples: FILE, REGISTRY, and DESKTOP.
 - **Policy** – This class identifies one, or a group of agent computers. Typically, instances of Policy classes are connected either to an instance of another Policy class, or instances of Configuration classes.
- 3 To modify the definition of an attribute, click its name in the Attribute List to highlight it. The name, length, and description of the highlighted attribute appear in the Attribute Information text boxes. The Type indicates the type of the highlighted attribute.
 - 4 Type your changes into the Attribute Information data entry fields, and click on the Type option to set the attribute's type. Class attributes can be one of four types as described in Table 41.

Table 41 Class Attribute types and their functions

Attribute Type	Usage
Expression 	Contains a single line REXX command that is executed during resolution. If the expression evaluates to “true,” in an attribute named ZSTOP, resolution of the current instance terminates
Variable 	A piece of named storage containing a variable value. The variable's value forms a part of the agent's resolved desired state, and can influence the resolution process through messaging or symbolic substitution.
Connection (available)	Class connections determine the path of resolution of an agent's desired state during the agent connect process. Available connections are attributes into which a connection to

Attribute Type	Usage
 <p>Connection (set)</p>  <p>Note: The letter in the icon denotes the connection type:</p> <ul style="list-style-type: none"> • A – Always • C – Class • I – Includes • R – Requires 	<p>another class can be set.</p> <p>Set connections are attributes containing a specification for a connection to another class.</p> <p>A class connection is, in effect, a branch in the resolution process. See Resolution Process in the <i>Essentials Guide</i>.</p> <ul style="list-style-type: none"> • An Always connection is always resolved, no matter what the name of its attribute. • A Class connection is resolved if its attribute's name is <code>_ALWAYS_</code>, or if the name of its attribute matches the current value of the system message. • An Includes connection is always resolved, no matter what the name of its attribute. • A Requires connection blocks resolution of instances in the required instance and its children whose instance names are identical to instance names in the requiring instance and its children. <p>Requires and Includes connection types are useful, among other things, for expressing inter-package relationships, connecting instances of the PACKAGE Class.</p> <p>When one package Requires another, components of the required package are superseded by the same component in the requiring package.</p> <p>When one package Includes another, both are deployed as a single entity.</p>
<p>Method</p> 	<p>Methods are programs executed as part of the resolution process. The method attribute identifies the program to be executed.</p>

As you make changes, they are immediately reflected in the Attribute List.

- 5 The **Properties** check boxes are either enabled or disabled depending upon the attribute type.

Agent Properties



All Client Automation agent properties, except for Protect and Execute, are reserved for future enhancements, and you should leave them set to their default values.

- Select the Agent Protect property to indicate that the value of an attribute stored in an object on the agent computer will be encrypted.
- Select the Agent Execute property, which applies only to method attributes, to indicate that the method to be executed will be executed on the agent computer.

Configuration Server Properties

The following table summarizes the significance of the Configuration Server properties.

- **Global**
When selected for a variable in a transient class, this indicates that during resolution the variable will be allowed to migrate (flow) from the child object to a parent object, prior to the transient object being resolved.
- **Default**
Used internally by Client Automation and is not available to Administrators. It prevents default values in transient-class objects from being incorporated into parent, persistent-class objects.
- **Substitute**
Indicates that symbolic substitution will occur for symbols (references to other variables identified by an initial ampersand) in the value of the variable. Clearing this check box treats the value of the variable as literal text, preventing symbolic substitution.

The Substitute property determines whether symbolic substitution occurs when a variable is created as a result of a Configuration Server GET operation, such as occurs when a REXX program calls the EDMGET function. If a variable in the object retrieved by EDMGET contains a symbol, and the Substitute check box was selected for that variable when the object was created, substitution occurs as the variable is created in storage.

For example, the expression `&(ZMASTER.ZUSERID)` will evaluate to WILLIAM after symbolic substitution, if the value of the ZUSERID attribute of the ZMASTER object is WILLIAM at the time symbolic substitution occurs.

- **Protect**
Indicates that the value of the attribute is to be encrypted during resolution. The attribute value exists in encrypted form in memory during resolution, but remains as plain text in the CSDB.
- **Absolute**
When selected for a variable in a persistent class, prevents identically named variables in child objects (instantiated from a transient class) from overlaying the values for those variables in the parent object.
- **Execute**
Indicates that the value of the attribute identifies a method to be executed by the Configuration Server.
- **Resolve**
Indicates that symbolic substitution will occur for symbols (references to other variables identified by an initial ampersand) in the value of the variable. Clearing this property's check box causes the value of the variable to be treated as literal text, preventing symbolic substitution.

The Resolve property determines whether symbolic substitution occurs when a variable is PUT into an object, such as occurs when a class instance is retrieved from the CSDB and instantiated as an object in storage.
- **Counter**
Indicates that the variable contains an integer that the Configuration Server will accumulate. The accumulated total will be stored in a variable with the same name in all parent persistent objects.

For example, this property can be used for calculating the total uncompressed size of all resources distributed for a service, and storing the result in the ZSERVICE object. Select this check box for the ZRSCSIZE variable in the FILE Class, and a ZRSCSIZE variable will be added to the ZSERVICE object, containing the sum of the ZRSCSIZE values of all FILE instances in the service. The total size of all services' resources for the user will automatically be stored in a ZRSCSIZE variable in the user's ZMASTER object. The ZRSCSIZE variables in the ZSERVICE and ZMASTER objects are created and maintained automatically by the Configuration Server when the Counter check box is selected for the ZRSCSIZE variable in the FILE Class template.

Any variable containing an integer can be accumulated and stored in parent persistent objects by selecting the Counter check box in the variable's definition in the class template. For example, select this check box on the ZCMPSIZE variable of the FILE Class to calculate the total compressed size of resources distributed for a service.

You can reset the property settings for the selected (i.e., highlighted) attribute to their default values by clicking the default buttons at the bottom of the Properties group box.

To add an attribute to a class

- 1 To add a new attribute, first determine where in the Attribute List the attribute should be inserted. Unless the Automatic Sequencing check box is selected, as described above, attributes are processed during resolution in the order in which they appear in the Attribute List.
- 2 Click an attribute next to the insertion location, to highlight it. Then click **Insert Before** or **Add After**. A blank attribute (a variable attribute by default) is inserted at the indicated location.
- 3 Enter a name, length, and description in the Attribute Information data entry fields, and then select the Type option from the drop-down list.

Attribute names can be up to eight characters long. Attribute names can have significance in the resolution process. Values of variables in child objects instantiated from within transient classes can replace the values of identically named variables in parent objects represented within persistent classes. Methods can be conditionally executed or skipped, and class connections can be conditionally followed or ignored based upon the name of the method or class connection attribute. If the attribute's name is `_ALWAYS_`, the method will be executed or the class connection will be followed unconditionally. Otherwise, the method will be executed, or a class connection will be followed only if the name of the attribute is identical to the current value of the system message. See the *Essentials Guide* for more information.

Specify the length of the attribute, in characters. Enter a number greater than or equal to the maximum number of characters for a value of the attribute. For connection and method attributes, a length of 50 is normally adequate. The length needed for a variable attribute depends on the data it is intended to hold.

The description you enter appears in the Attribute Description column of the Editing Instance dialog box. The clearer the description you enter now, the easier it will be for those who edit instances of this class later. For variable attributes, enter a description that clearly identifies the data to be held in the variable attribute. For class connections, you can enter Connect to, or a description that is more meaningful for your purposes. For methods and expressions, try to indicate the purpose of the method or expression in the description.

- To set the value of an attribute as the display name in the CSDB Editor, select the attribute, and then select the **Attribute Contains the Instance Alias Name** check box. This is usually set on the Name (friendly name) attribute. Only one attribute should have this turned on per instance.

Optional Attribute Editing Controls

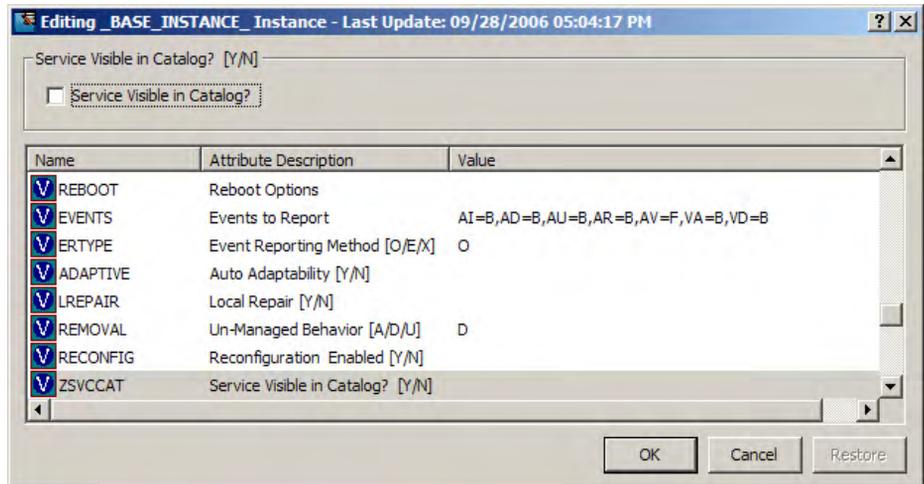
The Attribute Description for variable attributes can be used to determine the control that the CSDB Editor displays for entering or editing that attribute in the Editing Instance dialog box. Normally, variable attributes are entered or edited as free text in a text box. You can change this to a check box, a set of check boxes, or a drop-down list.

To enable this feature, select the Use Edit Masks When Editing Instances check box on the Instance Options tab of the Options dialog box. See [Table 38](#) on page 132 for details.

- **Flag Attribute**

A flag attribute is a one-character field whose value can be Y or N. To display a check box for entering or editing flag attributes in instances, the Attribute Description for the attribute in the class template must terminate with [Y/N], as in [Figure 17](#) below:

Figure 17 ZSVCCAT variable contains [Y/N] for flag attribute



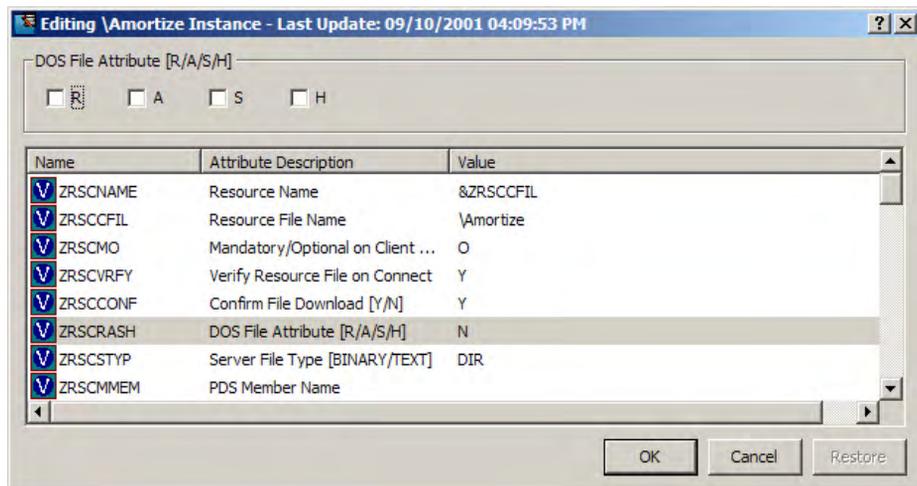
The ZSVCCAT attribute is one character in length, and its Attribute Description, Service Visible in Catalog? [Y/N], ends with [Y/N]. Thus, to enter or edit this variable, the CSDB Editor user interface contains a check box.

- **Flag Set Attribute**

A flag set attribute is a two- or more character field that holds a value composed of zero or more single-character flags. If a flag is on, the character value that is associated with the flag appears in the flag set string. If a flag is off, the character value that is associated with the flag does not appear in the flag set string. An empty flag set string indicates that no flags are on.

For example, the DOS File Attribute flag set for a particular file is contained in the ZRSCRASH attribute of the FILE Class. You use this four-character attribute to indicate which of the DOS File Attributes (R=Read-only, A=Archive, S=System, H=Hidden) should be turned on for the file associated with the FILE instance when that file is deployed to a user desktop. When editing this attribute in an instance, it appears as in Figure 18 below:

Figure 18 An attribute ending with a set of one-character choices



If the Attribute Description ends with one-character choices separated by slashes, and the number of choices equals the length of the attribute (as in the ZRSCRASH attribute, above), the CSDB Editor treats the attribute as a flag set. It displays a check box for each potential value in the flag set in the Editing Instance dialog box. If a check box is selected, its associated value is inserted in the attribute.

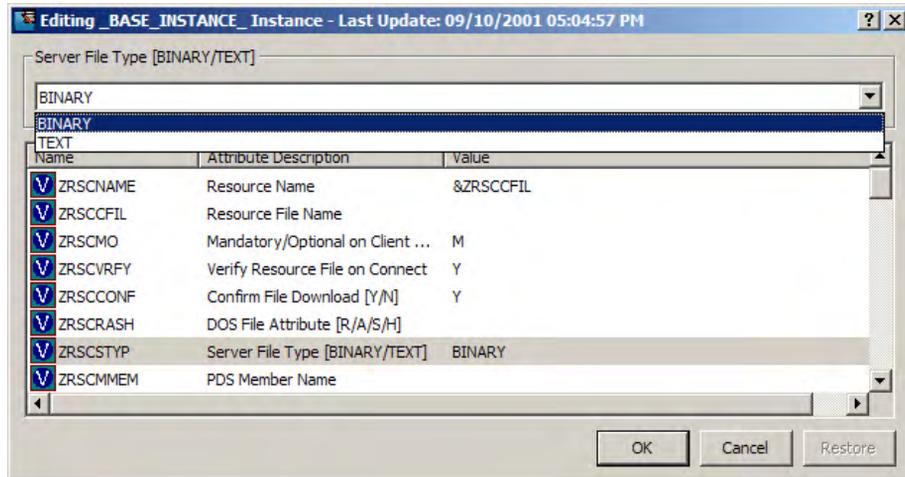
- **Drop-Down List**

The CSDB Editor displays a drop-down list for entering or editing attributes whose Attribute Description in the class template terminates

with a series of choices separated by slashes and enclosed in square brackets.

For example, [Figure 19](#) below shows that the ZRSCSTYP attribute of the FILE Class has two choices in the Attribute Description. An attribute description that terminates with a series of choices separated by slashes and enclosed in brackets will have a drop-down list for editing attributes.

Figure 19 ZRSCSTYP attribute



- 1 As you make changes to a class, you might change your mind and wish to start over. After a change has been made during class editing, the Restore button is enabled. Click this button to discard changes you have made to the class since the Editing Class dialog box opened.

The Confirm Restore message appears.

- Click **Yes** to discard changes and restart editing the class.
 - Click **No** to dismiss the confirmation without discarding your changes.
- 2 When you are finished editing the class, click **OK** to save your changes and exit the Editing Class dialog box.
 - If there are blank attribute names or descriptions in the class, or if an attribute's length is zero, an alert message opens.
 - If there are no blank attribute names or descriptions in the class, and no attribute's length is zero, the Class Edit Confirmation message opens. It warns you to back up your database before proceeding.

- 3 Click **Yes** to update all existing instances of the class.
Click **No** to discard your changes to the class, and avoid updating the instances.
Click **Cancel** to return to the Editing Class dialog box.

Deleting a Class

To delete a class and all of its instances

- 1 Right-click on the class in the CSDB Editor tree view.
- 2 Select **Delete Class** from the shortcut menu that opens. A Confirm Delete dialog box opens.
- 3 Click **OK** to delete the class and all of its instances.
Click **Cancel** to close the message without deleting anything.

Working with Instances

Much of the work of administrators consists of adding, editing, and deleting instances of various classes, and setting or removing connections between them.

Naming Instances

Use these guidelines when naming instances in the CSDB:

- Use characters that can be displayed and printed.
- Do not include a period (.). Periods are reserved characters used to delimit the File, Domain, Class, and Instance names (F.D.C.I) of a fully-qualified instance in the CSDB.
- Do not include special characters that have a specific meaning to the file system or command shell of any operating system, such as these:
 \ | / : ; < > * ? ` " ' , () &
- Do not include sequential underscores. This reserved sequence indicates a Group Resource in the CSDB.
- Do not include embedded spaces or control characters, such as blanks, CR (Carriage Return), LR (Line Return), or tabs.

Friendly Instance Names

Friendly instance names do not have the same restrictions as instance names. Use characters that can be displayed and printed. Spaces and other

special characters are allowed. To avoid confusion, it is recommended that you do not use periods in friendly names.

Adding an Instance

You can add a new instance from scratch or copy an existing instance. Which method you choose depends upon whether an instance exists whose attribute values are substantially similar to the instance you want to add. If so, copy the existing instance; otherwise, add a new instance from scratch.

Copying an Instance

Copying an instance creates a new instance with the same attribute values as the copied instance.

To copy an existing instance

- 1 Choose **Copy Instance** from a shortcut menu. A Copy Instance dialog box opens.
- 2 Enter a display name (i.e., a friendly name) and an instance name for the new instance, and click **OK**.

The instance will be created and will appear in both the tree view and in the list view of the CSDB Editor window.

The friendly name appears in the tree view, and the instance (i.e., internal) name appears in the title bar of the list view of the CSDB Editor window.

Adding an Instance

Creating a new instance results in an instance with attribute values set as in the `_BASE_INSTANCE_` of the class.

To add a new instance

- 1 Choose **New Instance** from a shortcut menu. A Create Instance dialog box opens.
- 2 Type a display name (i.e., a friendly name) and an instance name, and click **OK**.

The instance will be created and will appear in both the tree view and in the list view of the CSDB Editor window.

The friendly name appears in the tree view, and the instance (i.e., internal) name appears in the title bar of the list view of the CSDB Editor window.

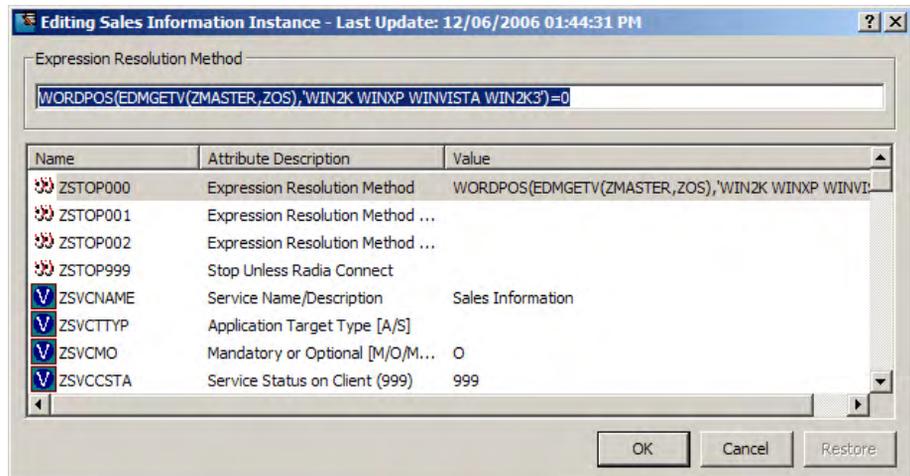
You can now edit the attribute value of the new instance.

Editing an Instance

This section describes how to edit a single instance.

To edit an instance

- 1 Choose **Edit Instance** from a shortcut menu, or double-click an attribute's name in the list view of the CSDB Editor window. The Editing Instance dialog box opens.



The dialog box lists the attributes in the instance, and their current values. The highlighted attribute is open for editing in the data entry field at the top of the dialog box.

- 2 Type the value into the text box at the top of the dialog box.

See [Optional Attribute Editing Controls](#) on page 152 for information about how the CSDB Editor determines which edit control to use for a particular attribute.

- 3 To highlight and select a different attribute, click its name.
- 4 Continue to edit each attribute you want to change.
- 5 Click **OK** to close the dialog box and save your changes.



If you selected the Prompt: Before Instance Updates check box on the Instance Options tab of the Options dialog box (see [Figure 11](#) on page 132), An Instance Edit Confirmation message box opens

- 6 If the Instance Edit Confirmation message box opens:
 - Click **Yes** to complete the update of the instance.
 - Click **No** to discard your changes to the instance, and return to the CSDB Editor main window.
 - Click **Cancel** to return to the Editing Instance dialog box.

When you edit connection attributes, notice the warning about editing class connections manually at the bottom of the dialog box. Connection values can be complicated to type, and a single typographical error can produce unexpected and undesirable results during resolution. It is recommended that you use the drag-and-drop method.

Editing Multiple Instances

The CSDB Editor enables you to edit multiple instances in a single operation.

One of the selected instances serves as a template. The template instance's values appear in the Editing Instance dialog box. When you change a value in the template instance, the change is applied to all the instances in the selected set. No changes are stored in the CSDB until you exit the dialog box and confirm the change operation.

To edit multiple instances at the same time

- 1 Select the instances in the list view of the CSDB Editor window.

Click on an instance to select it.

 - **Ctrl**-click to add an instance to the selected set

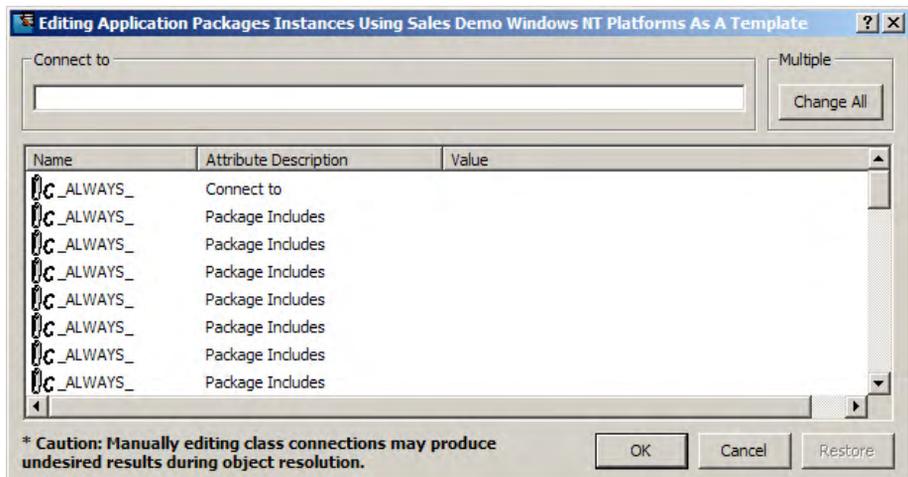
or

 - **Shift**-click and/or **Ctrl-Shift**-click to add a range of instances to the selected set.

Selected instances are highlighted:

Name	Instance Name	Type
_BASE_INSTANCE_	_BASE_INSTANCE_	SOFTWARE.PACKAGE Instance
_CLASS_BEHAVIORS_FILE_	_CLASS_BEHAVIORS_FILE_	SOFTWARE.PACKAGE Instance
_CLASS_BEHAVIORS_REGISTRY_	_CLASS_BEHAVIORS_REGISTRY_	SOFTWARE.PACKAGE Instance
_CLIENT_BEHAVIORS_	_CLIENT_BEHAVIORS_	SOFTWARE.PACKAGE Instance
Amortize Windows NT Platforms	AMORTIZE2_NT	SOFTWARE.PACKAGE Instance
Drag & View Windows NT Platforms	DRAGVIEW2_NT	SOFTWARE.PACKAGE Instance
GS-Calc Windows NT Platforms	GSCALC2_NT	SOFTWARE.PACKAGE Instance
Radia Behaviors 5/18/99	ADAPT	SOFTWARE.PACKAGE Instance
Radia Behaviors Panels	RADIA_BEHAVIORS_PANELS	SOFTWARE.PACKAGE Instance
Radia Client Behaviors	CLIENT_BEHAVIORS_237	SOFTWARE.PACKAGE Instance
Redbox Organizer Windows NT Platforms	REDBOX2_NT	SOFTWARE.PACKAGE Instance
Sales Demo Windows NT Platforms	SALES2_NT	SOFTWARE.PACKAGE Instance
Sales Update - Microsoft	SALES_MS	SOFTWARE.PACKAGE Instance
StratusPad Windows NT Platforms	STRATUSPAD2_NT	SOFTWARE.PACKAGE Instance
TightVNC	TIGHTVNC	SOFTWARE.PACKAGE Instance
Update and Install Behavior August 10, 1...	CLIENT_BEHAVIOR_081098	SOFTWARE.PACKAGE Instance

- 2 Right-click on the instance that you want to use as the template for the multiple instance-edit operation. A shortcut menu opens.
- 3 Select **Edit Multiple Instances**. The Editing Instance dialog box appears:

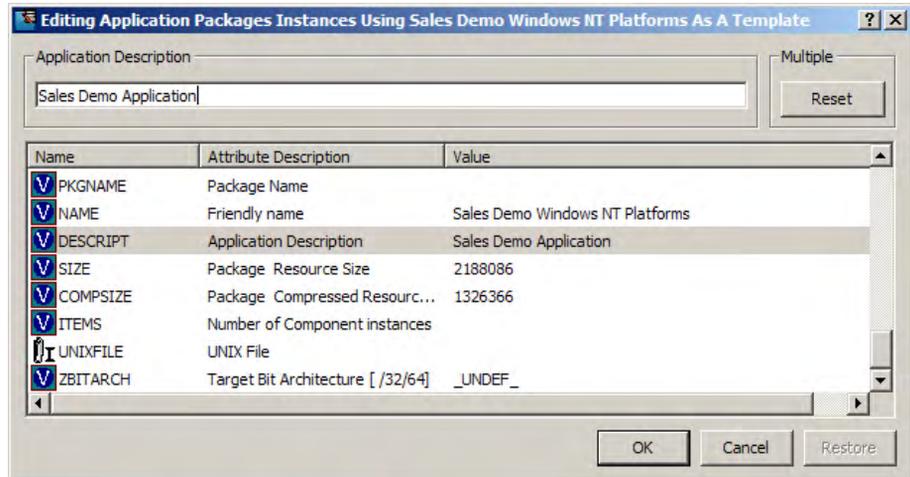


The title bar displays how many instances are in the set of selected instances, the selected instances' class friendly name, and the friendly name of the instance serving as the template.

- 4 Scroll to and select a variable whose value you want to copy to all instances in the selected set.

- If the selected variable in the template instance already has the desired value, click **Change All** to copy that value to the other instances in the selected set.
- If the selected variable does not have the desired value, edit the value.

In either case, the caption of the Change All button will change to Reset.



To restore the value of a currently selected variable to its original contents:

- Click **Reset**.
As you scroll through the instance variables, you can determine if you made a change to the selected variable by referring to the caption on this button.
 - If the caption is Change All, you have made no change to the variable during the current editing session.
 - If the caption is Reset, you have made a change to the selected variable.

After making a change to at least one variable, and selecting a different variable, the Restore button becomes enabled.

To undo all changes made in the current editing session

- 1 Click **Restore**. The Confirm Restore message opens.
- 2 Click **Yes** to discard all changes made in the current editing session.
Click **No** to resume the editing session without restoring original values.

Changes made during the editing session are not saved in the CSDB until you click **OK** and confirm your changes.

- 3 Click **OK** to save the changes made in the template instance to all of the selected instances.

Click **Cancel** to resume the Editing Instance dialog box session.

You can click **Cancel** in the Editing Instance dialog box to exit the dialog box without saving any changes.

Renaming an Instance

Be careful when renaming instances. Instances are connected via the connection attributes they contain. When you rename an instance, any other instance connected to the renamed instance will have a broken connection. This can produce unexpected and undesirable effects during resolution. Before renaming an instance, make sure you know which other instances are connected to the renamed instance, and edit them appropriately.

To rename an instance

- 1 In the tree view right-click the instance. A shortcut menu opens.
- 2 Select **Rename Instance**. A Rename Instance dialog box opens.
- 3 Enter a new display name (that is, a friendly name) and/or a new instance name for the instance, and click **OK**.

Deleting an Instance

Be careful when deleting instances. Instances are connected via the connection attributes they contain. When you delete an instance, any other instance connected to the deleted instance will have a broken connection. This can produce unexpected and undesirable effects during resolution. Before deleting an instance, make sure you know which other instances are connected to the instance to be deleted, and edit them appropriately.

To delete an instance

- 1 Right-click on the instance. A shortcut menu opens.
- 2 Select **Delete Instance**. The Confirm Delete message opens.
- 3 Click **Yes** to delete the instance.
Click **No** to close the dialog box without deleting the instance.

Password Authentication

This section describes how to use **EDMSIGN** and **EDMSIGNR** to enable password authentication when the Client Automation agent and Administrator log on to the Configuration Server.

EDMSIGN and EDMSIGNR

Use one of two ZMETHOD instances, SIGNON (which uses EDMSIGN) and RACF_SIGNON (which uses EDMSIGNR), to enable security configurations.

Although the RACF_SIGNON instance might appear to implement only RACF security, this method can be used to access other platform-specific security adapters.

EDMSIGN

EDMSIGN enables the Configuration Server to use the ZPWD variable for password authentication.

- When a Client Automation agent connects to the Configuration Server, the value of ZPWD in the ZMASTER object is compared with that which is stored in the user's profile.
- When the Administrator connects to the Configuration Server, the value of ZPWD in the ZADMIN object is compared with that which is stored in the user's profile.

EDMSIGNR

EDMSIGNR enables the Configuration Server to use an external security system such as Windows security for password authentication when a Client Automation agent or Administrator initiates a session with it.

 **Windows Users' Notes:** The user credentials that are submitted to this method *must* be valid on the machine that is running the Configuration Server—not the Client Automation agent or Administrator machine. Ensure that the user has been given all the necessary rights needed to execute applications and run services, based on the type of Windows security being used.

Be sure to distinguish between the two types of Windows security: Workgroup Security and Domain Security.

- **Windows Workgroup Security:** rights for specific user IDs (SIDs) are evolved from

Workgroup_Name + Computer_Name + Group_User_Belongs_To + User_Name.

- **Windows Domain Security:** user rights are evolved from Domain_Name + Domain_Group_User_Belongs_To + Computer_Name + Computer_Group_User_Belongs_To + User_Name.

The password that is stored in the ZPWD variable in the ZMASTER (for a Client Automation agent) or ZADMIN (for an Administrator) object is compared with the object that is stored in the native security system for that user ID.

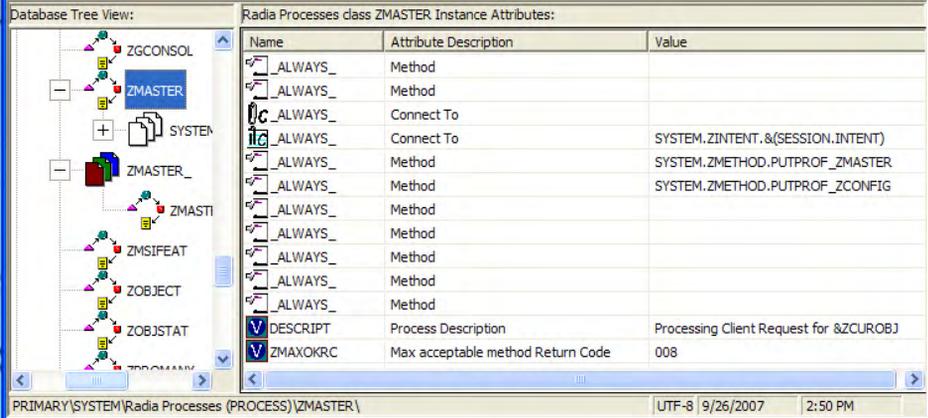
Without Password Authentication Enabled

This section describes how the Configuration Server determines that no password is required for log on.

Agent

Password authentication is *not* enabled for a Client Automation agent when the first method attribute in SYSTEM.PROCESS.ZMASTER is blank as shown in Figure 20 below.

Figure 20 Password authentication is not enabled



Name	Attribute Description	Value
ALWAYS	Method	
ALWAYS	Method	
ALWAYS	Connect To	
ALWAYS	Connect To	SYSTEM.ZINTENT.&(SESSION.INTENT)
ALWAYS	Method	SYSTEM.ZMETHOD.PUTPROF_ZMASTER
ALWAYS	Method	SYSTEM.ZMETHOD.PUTPROF_ZCONFIG
ALWAYS	Method	
DESCRIPT	Process Description	Processing Client Request for &ZCUIOBJ
ZMAXOKRC	Max acceptable method Return Code	008

In this case, when a Client Automation agent connects to the Configuration Server:

- 1 The Client Automation agent sends its ZMASTER object to the Configuration Server.

- 2 The Configuration Server looks at the first `_ALWAYS_` method attribute in `SYSTEM.PROCESS.ZMASTER` instance.
- 3 If this attribute is blank, no password is required, and log on is allowed.

Administrator

When the Administrator connects, it sends its `ZADMIN` object to the Configuration Server. In this case, the Configuration Server looks at the `SYSTEM.PROCESS.ZADMIN` instance. If this value is blank, then no password is required.

With Password Authentication Enabled

This section describes how the Configuration Server determines that a password *is* required.

Agent

When an agent attempts to connect to the Configuration Server, the Configuration Server looks at the `SYSTEM.PROCESS.ZMASTER` instance. If the first method attribute in `SYSTEM.PROCESS.ZMASTER` contains a value, e.g., `SYSTEM.ZMETHOD.AGENT_SIGNON` as shown in [Figure 21](#) below, the Configuration Server then looks up the method to determine if a password is required for log on.

Figure 21 Password authentication is enabled

Name	Attribute Description	Value
<input type="checkbox"/> _ALWAYS_	Method	AGENT_SIGNON
<input type="checkbox"/> _ALWAYS_	Method	
<input type="checkbox"/> _ALWAYS_	Connect To	
<input type="checkbox"/> _ALWAYS_	Connect To	SYSTEM.ZINTENT.&(SESSION.INTENT)
<input type="checkbox"/> _ALWAYS_	Method	SYSTEM.ZMETHOD.PUTPROF_ZMASTER
<input type="checkbox"/> _ALWAYS_	Method	SYSTEM.ZMETHOD.PUTPROF_ZCONFIG
<input type="checkbox"/> _ALWAYS_	Method	
<input checked="" type="checkbox"/> DESCRIPT	Process Description	Processing Client Request for &ZCUIOBJ
<input checked="" type="checkbox"/> ZMAXOKRC	Max acceptable method Return Code	008

Database Tree View: ZERROR, ZGCONSOL, ZMASTER, ZMASTER_, ZMSIFEAT, ZOBJECT, ZOBJSTAT, ZPROMANY, ZREPORT

PRIMARY\SYSTEM\Radia Processes (PROCESS)\ZMASTER\ UTF-8 | 9/26/2007 | 3:20 PM

In this case, when a Client Automation agent connects to the Configuration Server.

- 1 The Client Automation agent sends its ZMASTER object to the Configuration Server.
- 2 The Configuration Server looks at the first `_ALWAYS_` method attribute in `SYSTEM.PROCESS.ZMASTER` instance. In the case shown in [Figure 21](#) on page 164, it finds the method `AGENT_SIGNON`.
- 3 The Configuration Server then locates this method instance in `SYSTEM.ZMETHOD` and retrieves the name of the method (from `ZMTHNAME` attribute) to use for authentication (in this case, either `EDMSIGN` or `EDMSIGNR`), as shown in [Figure 22](#) below.

Figure 22 `SYSTEM.ZMETHOD.SIGNON` with `EDMSIGN` method

Name	Attribute Description	Value
ZMTHPRMS	Parameters Passed to Method	&(ZCURRENT.ZCURINST)
ZMTHTYPE	Method Type [REXX/ASM/EXE]	ASM
ZMTHNAME	Member Name of Method	EDMSIGN
DESCRIPT	Method Description	Manager Method &ZMTHNAME
ZMTHMODE	Mode [INTERNAL] or [EXTERNAL]	EXTERNAL
ZMTHSYNC	Synchronization Flag [Y] [N]	Y
ZMTHDSC1	Method Description 1	Processing Authentication Request for User=&ZUSERID
ZMTHDSC2	Method Description 2	
ZMUSTRUN	Return Code critical to Resolution?	Y

- 4 If the method to use for authentication is `EDMSIGN`, the Configuration Server now retrieves the password from the `ZMASTER` object that it just received from the agent. If it matches the password in the `ZPWD` variable of the `ZMASTER` object for this agent in the database, log on is allowed. If not, log on is denied.

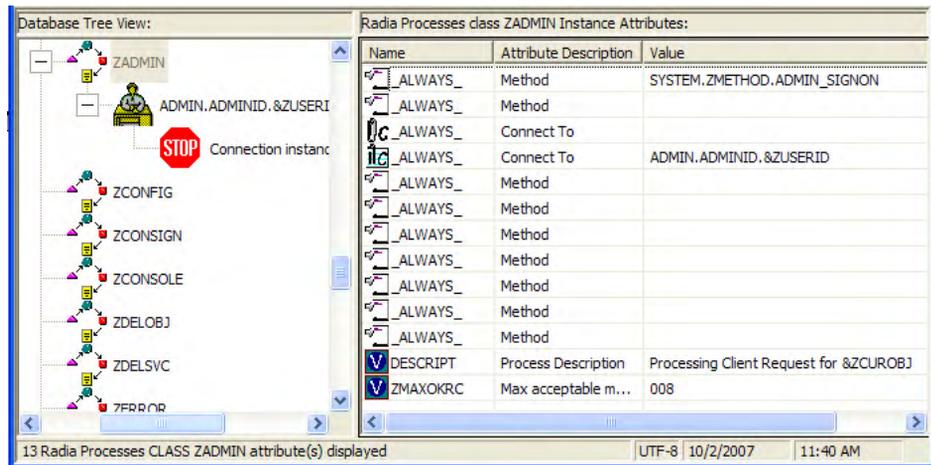
If the method to use for authentication is `EDMSIGNR`, the Configuration Server uses an external security system, such as Windows, for password authentication.

Administrator

When an Administrator attempts to connect to the Configuration Server, the Configuration Server looks at the `SYSTEM.PROCESS.ZADMIN` instance. If the first method attribute in `SYSTEM.PROCESS.ZADMIN` contains a value, e.g.,

SYSTEM.ZMETHOD.ADMIN_SIGNON as shown in Figure 23 below, the Configuration Server then looks up the method to determine if a password is required for log on.

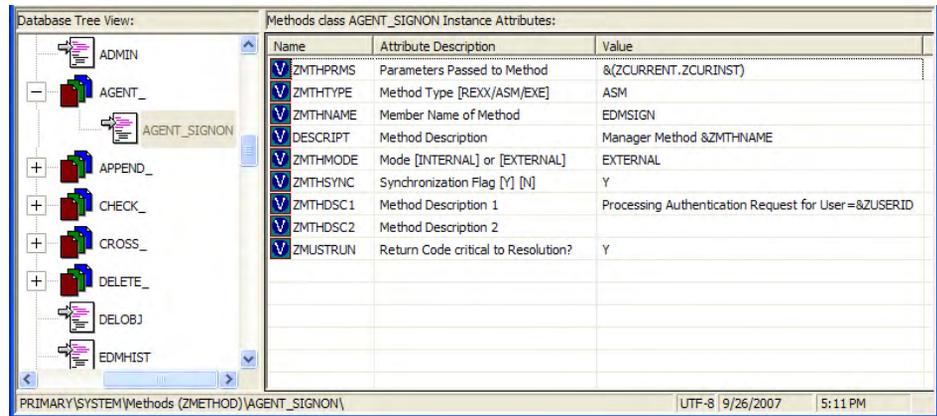
Figure 23 SYSTEM.PROCESS.ZADMIN



In this case, when the Administrator connects to the Configuration Server:

- 1 The Administrator sends its ZADMIN object to the Configuration Server.
- 2 The Configuration Server looks at the first `_ALWAYS_` method attribute in the `SYSTEM.PROCESS.ZADMIN` instance. In the case shown in Figure 23 above, it finds the value `SYSTEM.ZMETHOD.ADMIN_SIGNON`.
- 3 The Configuration Server then locates this method instance in `SYSTEM.ZMETHOD` and retrieves the name of the method (from `ZMTHNAME` attribute) to use for authentication (in this case, either `EDMSIGN` or `EDMSIGNR`).

Figure 24 SYSTEM.ZMETHOD.SIGNON with EDMSIGN method



- 4 If the method to use for authentication is EDMSIGN, the Configuration Server determines that a password is required for log on and retrieves it from the ZADMIN object that it just received from the Administrator. If it matches the password in the ZPWD variable of the ZADMIN object for this administrator in the database, log on is allowed. If not, log on is denied.

If the method to use for authentication is EDMSIGNR, the Configuration Server uses an external security system, such as Windows, for password authentication.

Enabling Password Authentication

This section contains instructions for enabling password authentication using EDMSIGN and EDMSIGNR. In the following examples, the Client Automation agent example uses EDMSIGN and the Administrator example uses EDMSIGNR. Both of these methods can be used with each of these Client Automation components.

To implement password authentication for Client Automation agents using EDMSIGN

- 1 Open the CSDB Editor and navigate the following path:
PRIMARY . SYSTEM . ZMETHOD.
- 2 In the ZMETHOD Class, locate the SIGNON instance, copy it, and re-name the copy (e.g., AGENT_SIGNON).

- 3 The new instance, **AGENT_SIGNON**, is now listed as an instance of the **ZMETHOD** Class.
- 4 Double-click **AGENT_SIGNON** to display its attributes in the list view of CSDB Editor.
- 5 Change the value of **ZMTHPRMS** to **ZMASTER**.
- 6 Navigate to the Radia Processes (**PROCESS**) Class of the **SYSTEM** Domain.
- 7 Double-click the **ZMASTER** instance to display its attributes in the list view of CSDB Editor.
- 8 Specify the value of the first **_ALWAYS_** (Method) field as **SYSTEM.ZMETHOD.AGENT_SIGNON**.

Password authentication has been implemented (using **EDMSIGN**) for Client Automation agents.

To implement password authentication for an Administrator using **EDMSIGNR**

- 1 Open the CSDB Editor and navigate the following path:
`PRIMARY.SYSTEM.ZMETHOD.`
- 2 In the **ZMETHOD** Class, locate the **RACF_** instance, copy it, and re-name the copy (e.g., **ADMIN_SIGNON**).
- 3 The new instance, **ADMIN_SIGNON**, is now listed as an instance of the **ZMETHOD** Class.
- 4 Double-click **ADMIN_SIGNON** to display its attributes in the list view of CSDB Editor.
- 5 Change the value of **ZMTHPRMS** to **ZADMIN**.
- 6 Navigate to the Radia Processes (**PROCESS**) Class of the **SYSTEM** Domain.
- 7 Double-click the **ZMASTER** instance to display its attributes in the list view of CSDB Editor.
- 8 Specify the value of the first **_ALWAYS_** (method) field as **SYSTEM.ZMETHOD.ADMIN_SIGNON**.



Be sure that the first **_ALWAYS_** (Connect To) field reads:
`ADMIN.ADMINID.&ZUSERID.`

Password authentication has been implemented (using **EDMSIGNR**) for an Administrator.

Changing Passwords

- ▶ The information in this section is applicable to changing *only* Client Automation passwords.

Client Automation passwords can be changed by either of the following methods.

- Open the Agent Explorer, In the ZMASTER object (for a Client Automation agent) or in the ZADMIN object (for an Administrator) specify a new password in ZNEWPWD and the old password in ZPWD.
- On the CSDB Editor log on screen, select the **Change Password** check box and specify the password information.

Access Rules and Administrators

An Administrator's database access can be defined as any of three levels of access, as detailed in [Table 42](#) below. CSDB Editor administrator access rules are stored in the ZACCESS Domain.

Table 42 Administrator Access-Level Values

Access Value	Definition
ALLOW	Configuration Server does <i>not</i> check the administrator access rules before granting access to an administrator. This value disables all administrator access-rule checks, even if they are defined in the database.
DENY	Configuration Server checks administrator access rules before granting access to an administrator. An administrator will be unable to perform an action unless there is an access rule defined in the database for that administrator specifically allowing it. This is the recommended setting when configuring administrator security. Notes: If this option is specified, undefined administrators will not have access to any CSDB administrator functions, unless the ADMINID._NULL_INSTANCE_ is modified to allow access to those functions.

Access Value	Definition
IGNORE	Configuration Server checks administrator access rules before granting access to an administrator. An administrator will be able to perform an action unless there is an access rule defined in the database for that administrator specifically prohibiting it.

All Configuration Servers have an edmprof file. This file is divided into multiple sections, and it defines the operational parameters for the Configuration Server, including its behavior regarding administrator access to the CSDB (as detailed in [Table 42](#) on page 169). The administrator access parameters are located in the MGR_ACCESS section.

To set/change the administrator access value

- 1 Shut down the Configuration Server.
- 2 Open the edmprof file.
- 3 Specify the ADMIN setting with one of the values in [Table 42](#) on page 169.
- 4 Save and close the edmprof file.
- 5 Restart the Configuration Server.

Creating Services with the New Application Wizard

After you create a package with the Packager or Publisher, use the New Application Wizard in the CSDB Editor to create a service. A **service** is the fundamental unit of content managed by Client Automation. Use the CSDB Editor to see services listed in the Application (ZSERVICE) class.

Task 1 Open the CSDB Editor

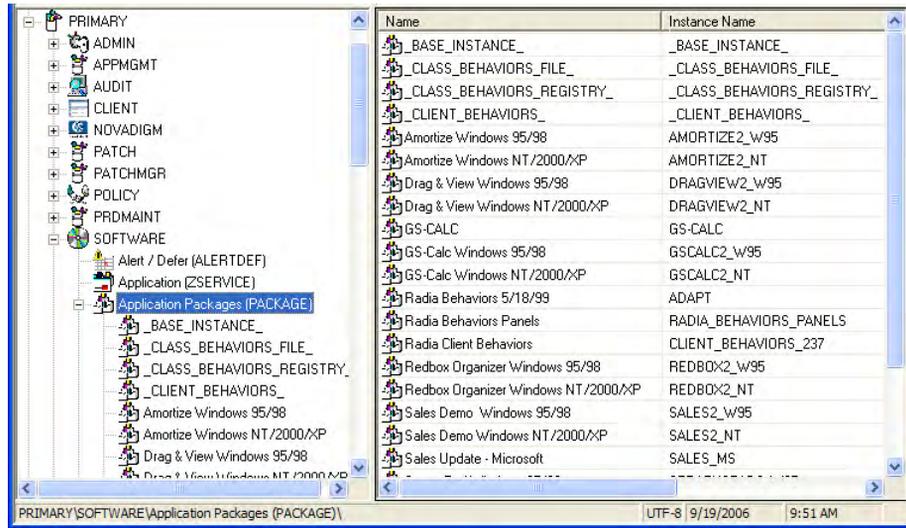
See [Getting Started](#) on page 125 if you need instructions for how to do this.

Task 2 Navigate to the Package Class

To navigate to the PACKAGE Class of the SOFTWARE Domain

- 1 Double-click **PRIMARY**. The domains of the PRIMARY File appear beneath its icon in the tree view and in the list view.

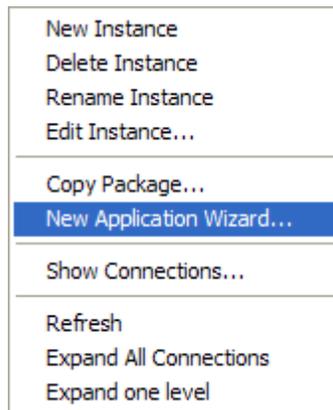
- 2 Double-click **SOFTWARE**. The classes of the SOFTWARE Domain appear beneath its icon in the tree view and in the list view.
- 3 Double-click **Application Packages (PACKAGE)** to open the PACKAGE Class. The instances of the PACKAGE Class appear beneath its icon in the tree view and in the list view as shown in the next image.



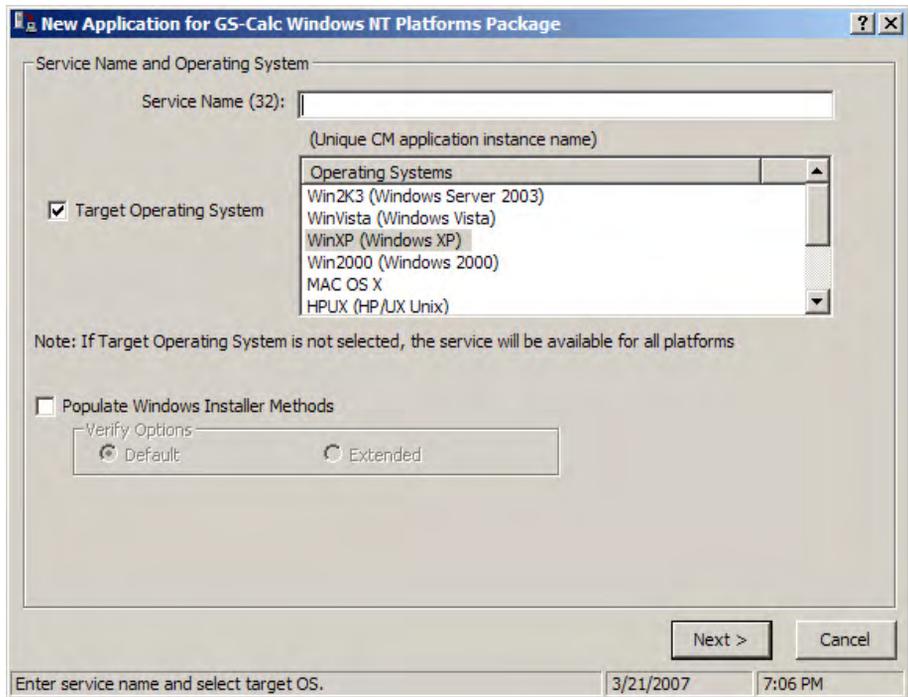
Task 3 Create a Service

To create a service using the New Application Wizard

- 1 In the PACKAGE Class of the SOFTWARE Domain, right-click the **PACKAGE Class** instance for which you want to create a service. A shortcut menu opens.

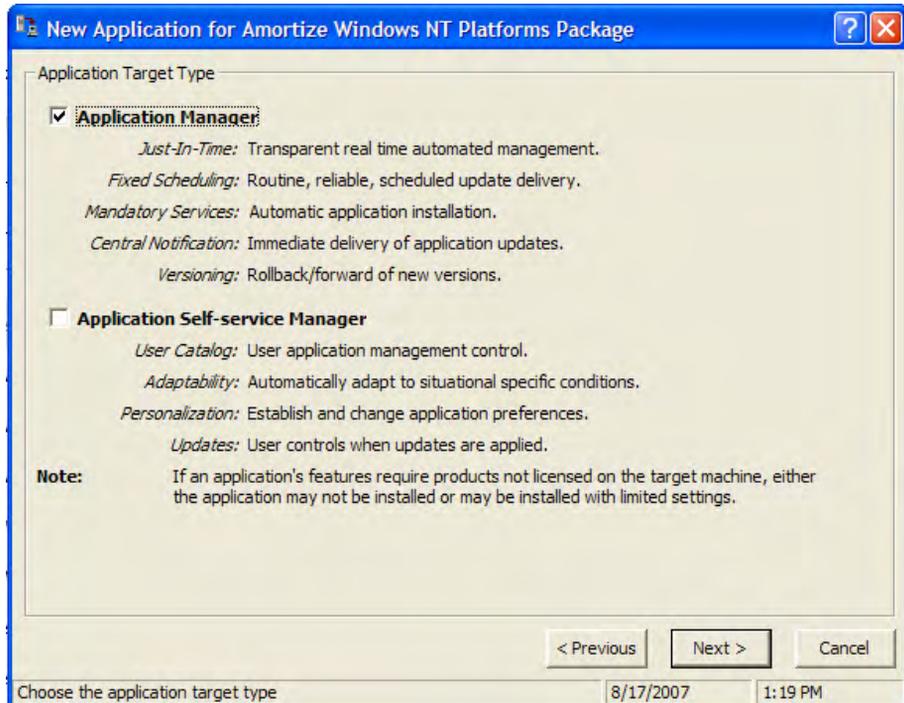


- 2 Click **New Application Wizard**. The New Application Wizard opens.



- 3 In the Service Name (32) text box, type a name, such as **WINZIP0009**, for the Application (ZSERVICE) instance. Follow the instance naming guidelines in [Working with Instances](#) on page 155.
- 4 Select the **Target Operating System** check box if you want to specify which operating systems are eligible for this service. Then, select the appropriate operating systems.

If you do not select the Target Operating System check box, the service is available on *all* platforms.
- 5 If you are creating a service for a Windows Installer-enabled application, you must select the **Populate Windows Installer Methods** check box.
- 6 Click **Next** to select the application target type.



- 7 Select the **Application Manager** check box. This designates the service as a mandatory application for your subscribers.

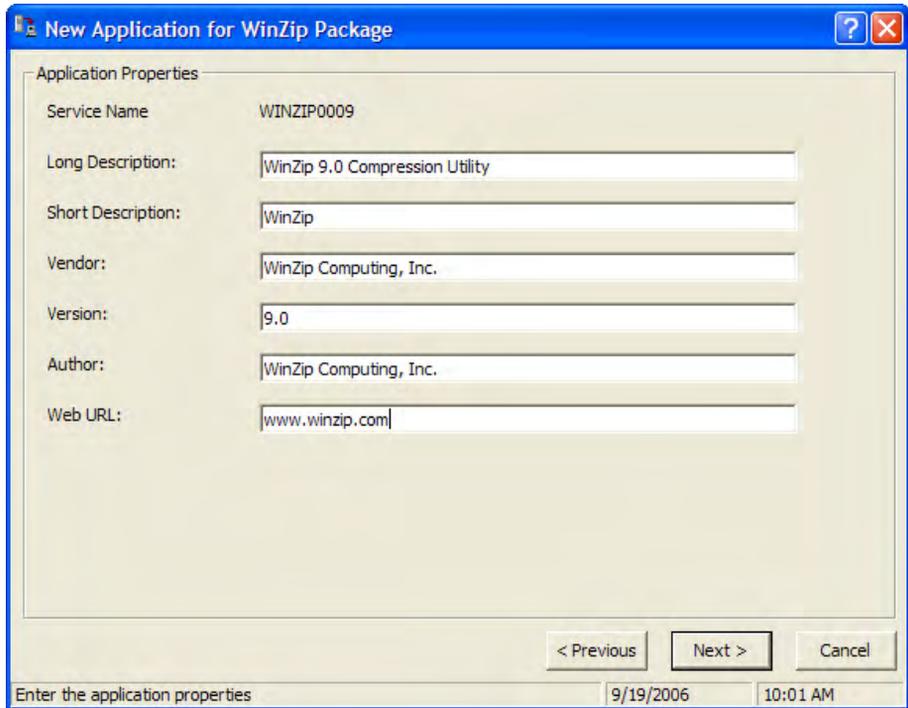


When using the Application Manager, applications *must* be mandatory in order to deploy them to your subscribers.

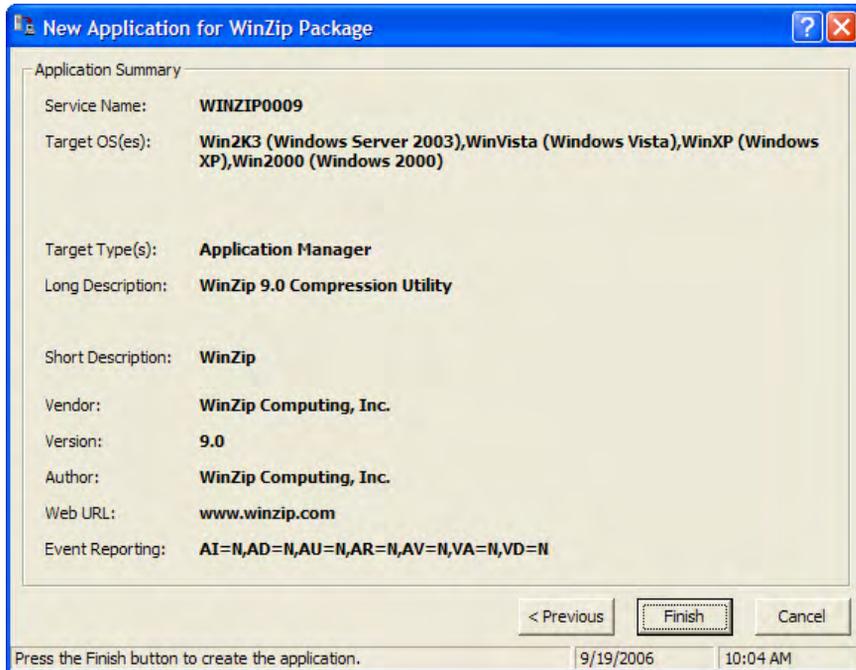
or

Select the **Application Self-service Manager** check box to designate this as an optional application that users may choose to install.

- 8 Click **Next** to enter the application properties.



- 9 Type the appropriate information in the text boxes as shown in the Application Properties dialog box.
- 10 Click **Next** to select the events that the Application Manager will report on.
- 11 Click the check box for each application event that you want to report on. Then, select the appropriate option button to indicate whether to report on the event's success, failure, or both.
or
Click **Use Base** if you want to inherit the values for the ERTYPE and EVENTS attributes from the base instance of the Application (ZSERVICE) instance. These attributes control event reporting.
- 12 If you want to save the current settings as the default settings for the Application Event Panel, click **Save as Default**.
- 13 Click **Next** to see a summary of your selections.



14 Click **Finish** to create the application instance.

15 Click **OK** to the message that the application has been added. The instance appears in the ZSERVICE Class.



Use the CSDB Editor to modify any of the information that you entered in the New Application Wizard. Locate the corresponding attribute in the ZSERVICE instance and change its value.

Refer to the *Application Manager and Application Self-service Manager Guide* for information on how to implement entitlement policy.

Defining MSI Basic Behaviors in the New Application Wizard

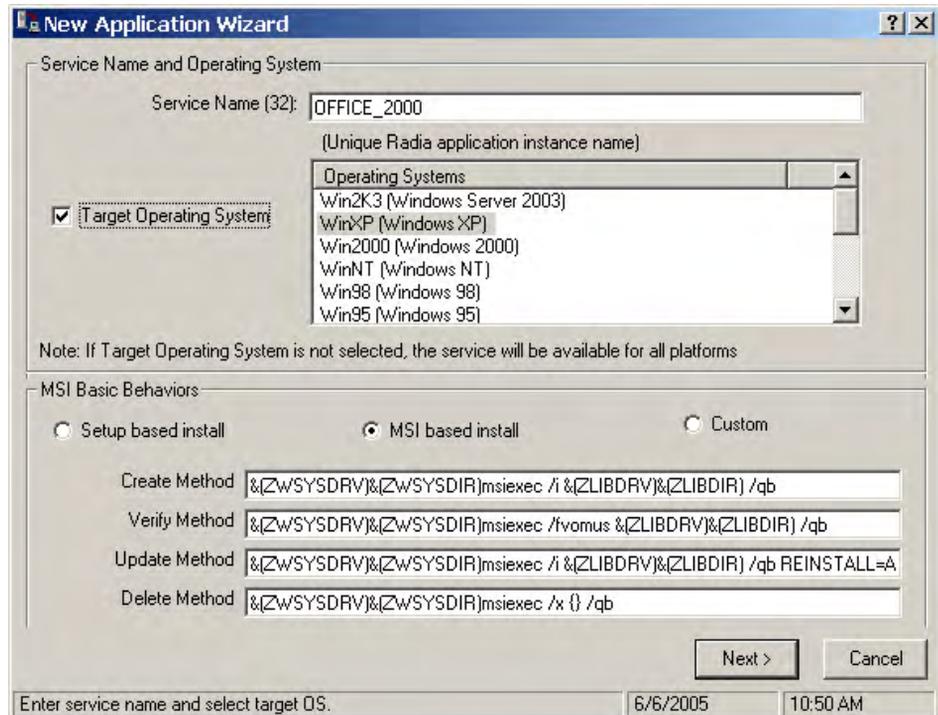
In contrast to Advanced Package Management, where Client Automation provides on-going management of the components of an MSI package, Basic Package Management is used for packages that:

- Do not support administrative installation.
- Are to be installed and uninstalled by Client Automation, but whose components are not to be managed by Client Automation. That is, the

Client Automation agent is not to be instrumental in any component verification or repair capability.

The New Application Wizard permits you to define MSI Basic Behavior methods for packages created with the Packager for Windows Installer. These packages have a connection to the MSIBASIC Class in the CSDB.

Figure 25 MSI Basic Behaviors in New Application Wizard



Packages that cannot be administratively installed will default to Basic Package Management with the Client Automation manageability features disabled. Use the MSI Basic Behaviors section of the New Installation Wizard dialog box to specify the install type and MSI native mode methods to be used for package management.

To define the MSI Basic Behaviors section of the New Application Wizard

- 1 Select the install type that reflects how the package is installed:

Setup-based install

The package installs with a `SETUP.EXE` file that calls `MSIEXEC` to install the package. This will run the native `setup.exe` for the MSI application.

If `setup.exe` supports any command line switches, such as `/s` for silent or `ADDLOCAL=ALL` for complete installation, you can add them to the Create Method. Not all MSI applications support the same command-line switches. For `setup.exe`, refer to the application's documentation for details.

MSI-based install

The package installs with `MSIEXEC`.

Custom

The package uses a custom installation method.

- 2 If necessary review and modify the native MSI commands and switches that are to be used for the Create, Verify, Update, and Delete Methods for basic package management.

The Version Group Editor

The **Version Group Editor** is a feature of CSDB Editor when the Application Manager is installed. Use it to create and edit instances of the **Version Group** Class. Features of the Version Group Editor enable you to create, edit, and delete Version instances of the Version Group and control the Version Group's deployment. Refer to the *Application Manager and Application Self-service Manager Guide*.

Service Optimization

Service Optimization uses patches to maximize the efficiency of distributing periodic updates to the packages in a service. Later packages contain revisions and additions to files published in earlier packages.

A **patch** is a published package of data (bytes) that is needed to upgrade or fix an application that is already published and installed on an agent computer. Since a patch contains only the differences between the packages, deployment of a patch takes less time and requires less network bandwidth. Patches should be built only for file changes that can be distributed more efficiently as patches than as replacement files.

Here's how Service Optimization works:

- 1 The Packager or the Publisher calculates the message-digest algorithm (**MD5**) signature of the file as it is being published. The MD5 algorithm can take input and create a 128-bit fingerprint (or message digest) of the

input. This information is then stored in compressed form in the Configuration Server. For more information about MD5 signature, visit:

www.faqs.org/rfcs/rfc1321.html

- 2 The PATCH Domain in the CSDB contains the list of available patches in the PATCH Class. The file signatures of a patch are used to identify the associated instance that contains the data for the patch. For more information about the PATCH Domain, refer to the *Configuration Server Database Reference Guide*.
 - ▶ Prior to deploying patches to users, change the ZOBJPTCH variable from N to Y in the USER Class of the POLICY Domain.
- 3 In the CSDB Editor, **Optimize Service** is an option on the Application (ZSERVICE) instance shortcut menu. It allows the administrator to create, display, and delete patches for selected application components.

Using the Optimize Service Option

In order to create a patch using the Optimize Service option, you first need to publish your packages using the Packager or the Publisher. For instructions on how to do this, see [Packager](#) on page 47 and [Publisher](#) on page 72.

- ▶ Patches can be created for packages that contain a signature only. Initially, only MD5 is supported.
Packages being used for patching must be published from the same location, or computer, to qualify for byte-level differencing patching. This will populate the eight-byte CRC found in the suffix of the instance names.
- ▶ Because the Packager and the Publisher calculate and use the MD5 signature, and this signature is vital to the patching functionality, there is no backward compatibility to previously published packages.

This section provides you with detailed instructions for:

- Creating an Application (ZSERVICE) in the SOFTWARE Domain for packages for which you will create a patch.
- Creating the patches using the Optimize Service wizard.

In this example, you will optimize a series of text files called BLD. These files were packaged and promoted to the CSDB. Each text file is its own package in the SOFTWARE Domain.

After you publish your packages to the CSDB, you need to create a ZSERVICE instance.



In some cases, the difference in size between two packages is so minimal that creating a patch will actually increase the size of the file. In this case, the creation of the patch will not be allowed.



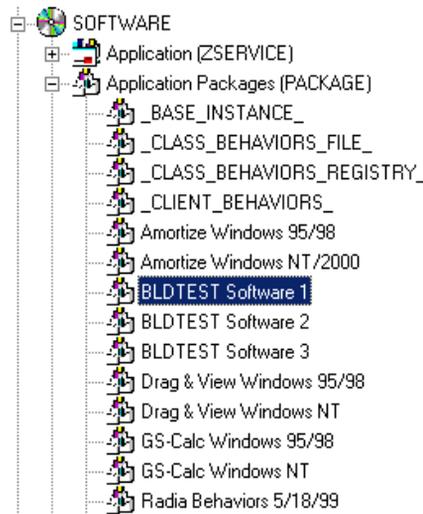
When publishing the files needed for this example, the following naming conventions were used:

- The file `bld1.txt` was originally published as `bld.txt`.
- When the second file was about to be published, it was renamed to `bld.txt`; the original `bld.txt` file was renamed `bld1.txt`.
- When the third file was about to be published, it was renamed `bld.txt`; the second file was renamed `bld2.txt`.

By using the same file name and the same machine during publishing, you are assuring the same CRC needed to correctly create patches.

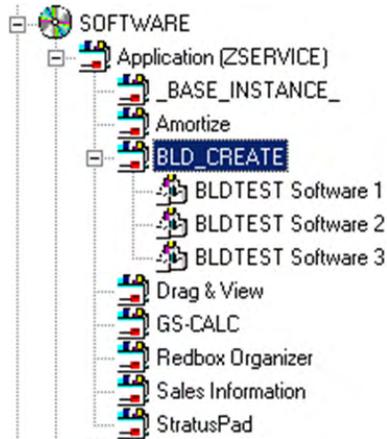
To configure the Optimize Service feature

- 1 Expand the tree view to the **PRIMARY.SOFTWARE.PACKAGE** Class and locate the **BLDTEST** packages.



- 2 Right-click on the **Application (ZSERVICE)** Class. A shortcut menu opens.
- 3 From the shortcut menu, select **New Instance**.

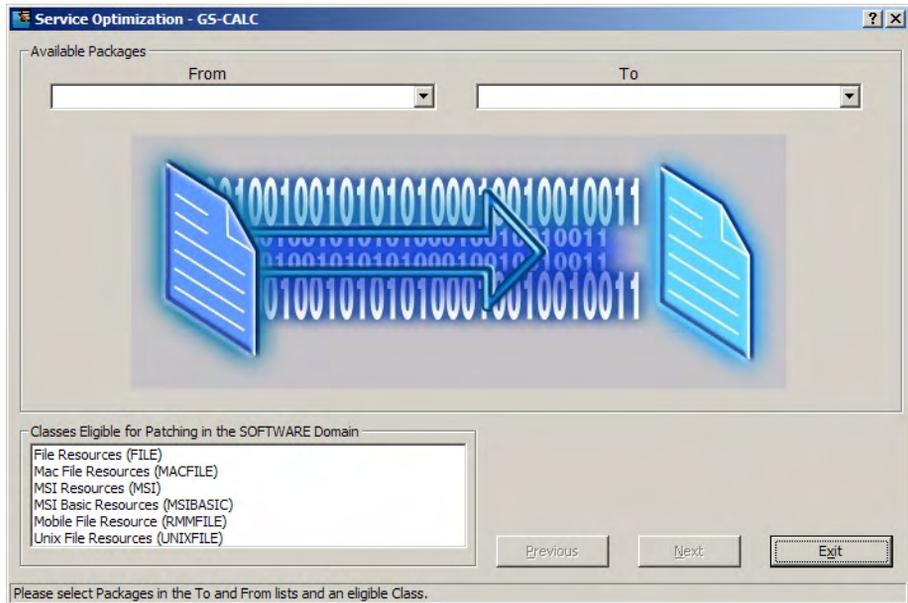
- 4 Create a new instance named BLD_CREATE, and click **OK**. The BLD_CREATE service is added to the CSDB.
- 5 Drag-and-drop the BLDTEST1, BLDTEST2, and BLDTEST3 packages to the BLD_CREATE instance in the Application (ZSERVICE) Class.



After you create the Application (in the ZSERVICE Class) that contains the Application Packages (from the PACKAGE Class), you can create your patches.

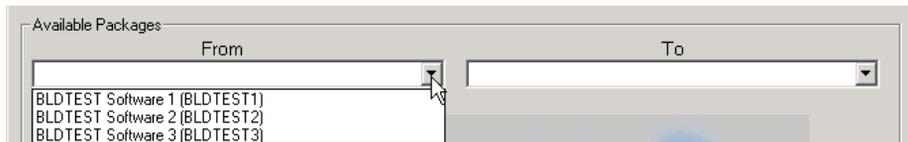
Just as you create a plan or policy for users, you might want to develop a plan for creating, editing, and deleting patches. Some important things to remember when creating patches:

- It is important to devise a means by which to document the packages that will create your patches. When building different patches in a small database, it is relatively easy to follow the schema of connected paths that create a patch. As your database grows, it is possible for the patch class, and the packages that they represent, to grow unwieldy.
 - Only those packages that contain an MD5 signature are eligible for patching.
 - Patching is different from *versioning*. If a patch fails to deploy properly, there is no previous version option.
- 6 Right-click the Application (ZSERVICE) Class **BLD_CREATE**. A shortcut menu opens.
 - 7 Select **Optimize Service**. The Service Optimization wizard opens.



- 8 Click the **Available Packages From** drop-down arrow. This lists the available packages for the service you are trying to optimize. When you click the **From** drop-down arrow, all of the packages for the Application (ZSERVICE) are displayed. In this example, click the **From** list to show all of the packages in the BLD_CREATE Application (ZSERVICE).

The packages that are available for the service you selected in step 6 BLD_CREATE, are listed.



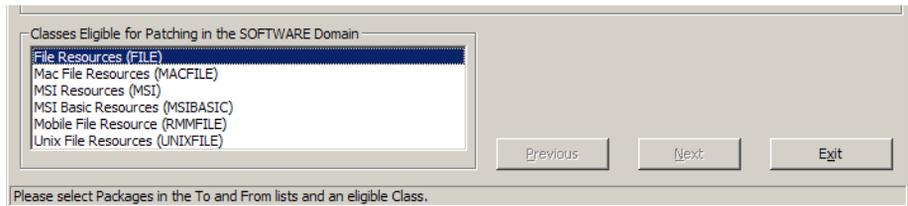
- 9 From the drop-down list select **BLDTEST Software 1 (BLDTEST1)**. The Available Packages To list is populated with the packages that are available for patching in the BLD_CREATE service.
- 10 Click the **Available Packages To** drop-down arrow. This lists those packages available for patching in the service you are optimizing.
- 11 From the drop-down list, select **BLDTEST Software 2 (BLDTEST2)**.



- 12 From Classes Eligible for Patching in the SOFTWARE Domain select the appropriate class for patching in the SOFTWARE Domain. Choose the class in which the file resources for the package are located.

➤ Currently, File Resources (FILE), MAC File Resources (MACFILE), MSI Resources (MSI), MSI Basic Resources (MSIBASIC), Mobile File Resources (RMMFILE), and Unix File Resources (UNIXFILE) qualify for patch building.

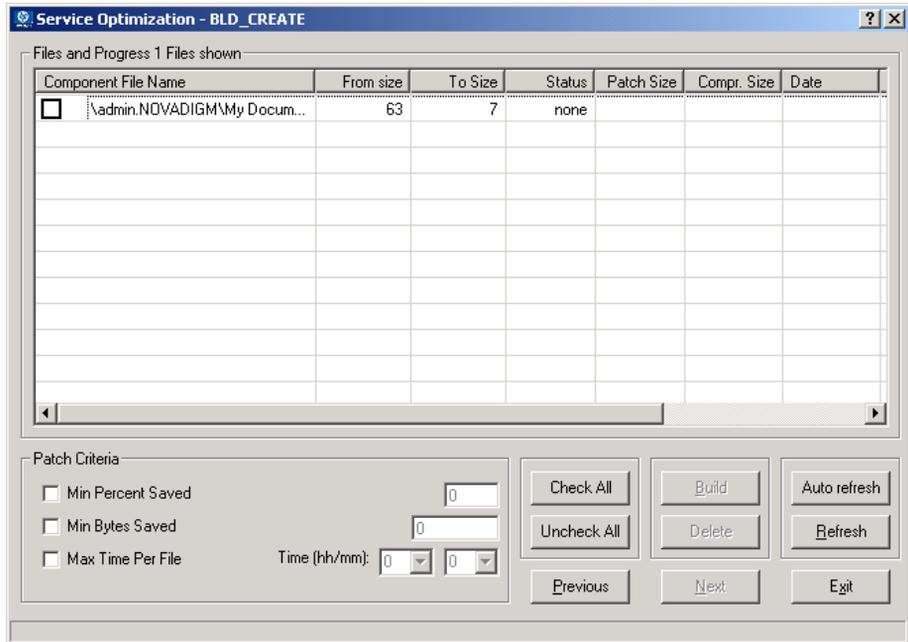
In this example, the only class that is available to be patched is File Resource (FILE).



➤ To determine which classes are available for patching within your package, expand your Application (ZSERVICE) instance to the attribute level. Examine your instances to determine which resources are available for patching.

- 13 Select **File Resource (FILE)** and click **Next** to continue.

The Files and Progress dialog box opens.



This dialog box displays files and progress information. The Service Optimization wizard gathers the instance information based on the selections made in the Available Packages dialog box.

Table 43 below shows the options in the Files and Progress section of this dialog box. And Table 44 on page 184 shows the options in the Patch Criteria area.

Table 43 Files and Progress information

Column	Description
Component File Name	The name of the patch that is being created.
From Size	The original resource-package size.
To Size	The original target-resource size.
Status	The status of the package. The valid values are none, queued, rejected, active, and built.
Patch Size	The size difference between the original package data (BLD1) and the changed package data (BLD2).
Compressed Size	The compressed size of the package listed in the From Size.

Column	Description
Date	The date on which the package was published.
Size Limit	This information is pulled from the Patch Criteria, Min Percent Saved field. This field is populated after the Patch has been built or when the patch building process has ended.
% Limit	This information is pulled from the Patch Criteria, Min Bytes Saved field. It is populated after the Patch has been built or when the patch building process is ended.
Time Limit	This information is pulled from the Patch Criteria, Max Time Per File field. This field is populated after the Patch has been built or when the patch building process is ended.

It is possible to set patch-building restrictions, such as percent size, bytes, and time elements by configuring the Patch Criteria options. You can control the creation of patches by setting criteria that will apply to all files whose check boxes are currently marked in the file list.

Suppose that the number of files in the list is large and you do not want to evaluate each file manually to determine if you want to build a patch for it. Click **Check All** to select all the files in the list. Then, use the Patch Criteria options to limit the building of patches to those files likely to produce a benefit.

[Table 44](#) below details the options of the Patch Criteria area. The default for these three options is an empty (unselected) check box.

Table 44 Patch Criteria information

File Option	Description
Min Percent Saved	Specify the minimum percent of the To component to save. Otherwise, the patch will not be built.
Min Bytes Saved	Specify the minimum number of bytes that the patch must save. Otherwise, the patch will not be built.
Max Time per File	Specify the maximum processing time for the patch. Use the drop-down list to indicate the maximum time allowable for patch building. The time can be set in minutes, or hours and minutes.

After a component file has been selected, the buttons used for patch building are enabled.

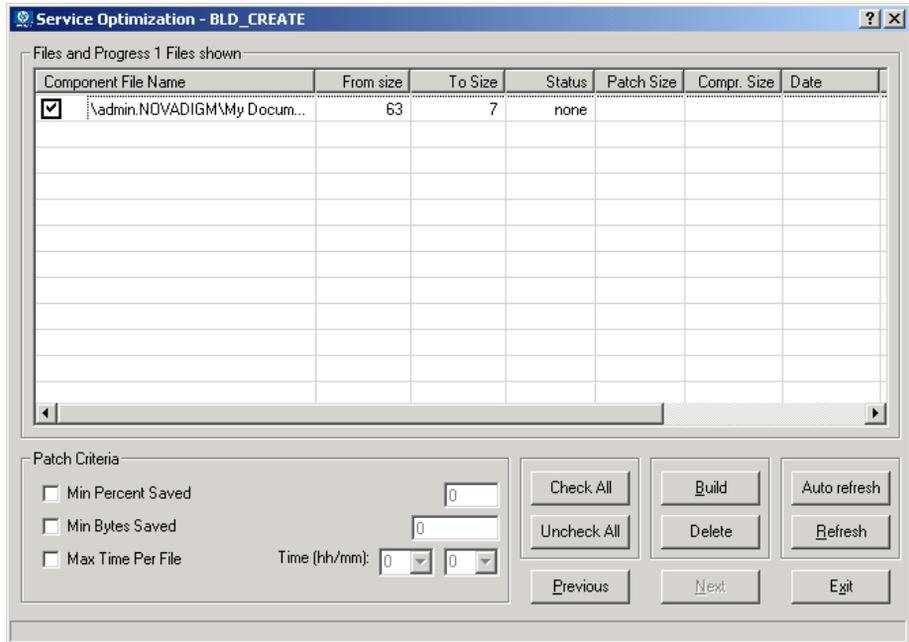
To apply a criterion, select its check box. The adjacent data entry fields are enabled. Enter the criteria into the associated data fields. The criteria for the patch are applied in combination. In order for a patch to be built for a selected file, it must meet all the criteria you apply.

Table 45 below explains the functions of the Patch Criteria buttons.

Table 45 Patch Criteria button functions

Button	Action
Check All	Selects all the files that are listed in the Component File Name column.
Build	Begins the patch build process.
Auto Refresh	Select Auto refresh to open the Auto Refresh Timer dialog box. When you use Auto Refresh, type a refresh interval between 1 and 99 minutes. You can cancel the auto refresh by typing 0.
Uncheck All	De-selects all files that are selected in the Component File Name column.
Delete	Deletes a selected component file.
Refresh	Refreshes build status.
Previous	Returns to the previous screen.
Next	Proceeds to the next screen.
Exit	Exits the Optimization Wizard.

14 Select the check box next to the files to be patched.



15 Click **Build**. The CSDB Editor will prompt you to confirm the new patch build.

16 Click **Yes** to continue.

or

Click **No** to exit.

When you click **Yes**, any current patches of the pair will be deleted. A patch-creation request for each selected file pair is then be sent to the Configuration Server.

17 To update the Status, click **Refresh**.

18 Click **Exit** to exit the Service Optimization wizard.

or

Click **Previous** to return to the Available Packages dialog box to build more patches.

The patch is located in the PATCH Domain, under the PATCH Class.

- A large product, MS Office for example, might need to be split into smaller sub-services to install only specific parts of the product suite.

Service grouping allows multiple services to be grouped by a higher, or **master**, service. This action allows administrators to logically group together similar or related components into one easy to understand collection.

Use the CSDB Editor to define and then group services and packages to form the master service.



Remember to target actions, such as notifies or timer updates, to the master service. Any action applied to the master service is applied to all sub-services.

Component Enhancements

- **Application Manager**
The RADSKMAN command contains the logic to process multiple service requests. It performs the request on each of the individual sub-services and reports the final results back to the master service.
- **Application Self-service Manager**
Each service in the Service List displays only the master service.

Prerequisites

Before you configure a master service for a service group, change or confirm the following settings:

- Configuration Server (versions 4.5 and above): in the `edmpprof` file, verify that `ALLOW_DUPLICATE_INSTANCES` in the `MGR_OBJECT_RESOLUTION` section is set to No.
- Configuration Server (versions prior to 4.5): in the `edmpprof` file, verify that in the `MGR_OBJECT_RESOLUTION` section:
 - `ALLOW_CIRCULAR_REFERENCE` is set to Yes
 - and
 - `ALLOW_DUPLICATE_INSTANCES` is set to No.
- CSDB

To configure a set of services (a grouping in which services depend on one another for installation), add a variable to the `_BASE_INSTANCE_` of the Application (ZSERVICE) Class. This is a variable called `ZSVCGRP`. If

you would like to add this variable to the ZSERVICE Class, use the following table to help you set the variable.

For information on editing a class, see [Editing a Class](#) on page 145.

Table 46 ZSVCGRP variable values

Value	Behavior
Y	Specify Y (Yes) for the Configuration Server to process the service groups as if they are unrelated applications or applications that are not dependent on one another for installation. If this variable is not present, this is the default.
D	Specify D (Dependent) for the Configuration Server to process the services under the service group as though they are dependent on one another for installation. If one sub-service fails, the remaining services will not be installed.
N	Specify N (No) for no service groups.

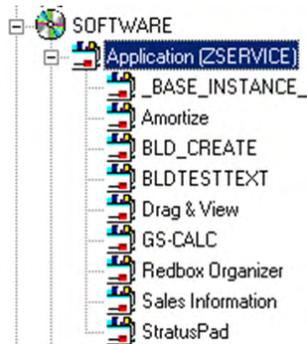
Creating a Master Service for a Service Group

After you make the changes to the `edmpprof` file, you can create a master service instance for a service group.

Use the CSDB Editor to create a new Application (ZSERVICE) instance for the master service. Next, for each service that is to be a member of the group, make a connection between the member service and the master service. You can add service members to the master service at any time.

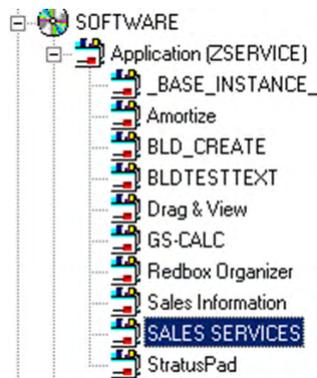
For our example, we will create a master service instance called Sales Services. This master service will contain the Amortize and GS-Calc services.

Figure 26 Sample services in the ZSERVICE Class



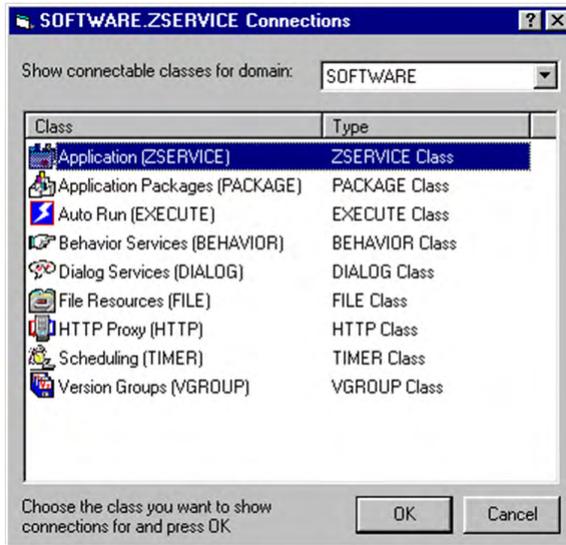
To create a Master Service for a Service Group

- 1 Open the CSDB Editor and navigate to the PRIMARY.SOFTWARE.ZSERVICE Class.
- 2 Expand the class to display the Application (ZSERVICE) instances.
- 3 Right-click **Application (ZSERVICE)**. A shortcut menu opens.
- 4 Select **New Instance** from the shortcut menu. The Create Instance dialog box opens.
- 5 Enter the name of the new ZSERVICE instance (SALES SERVICES).
- 6 Click **OK** to create the instance. The new instance appears in the Application (ZSERVICE) Class.

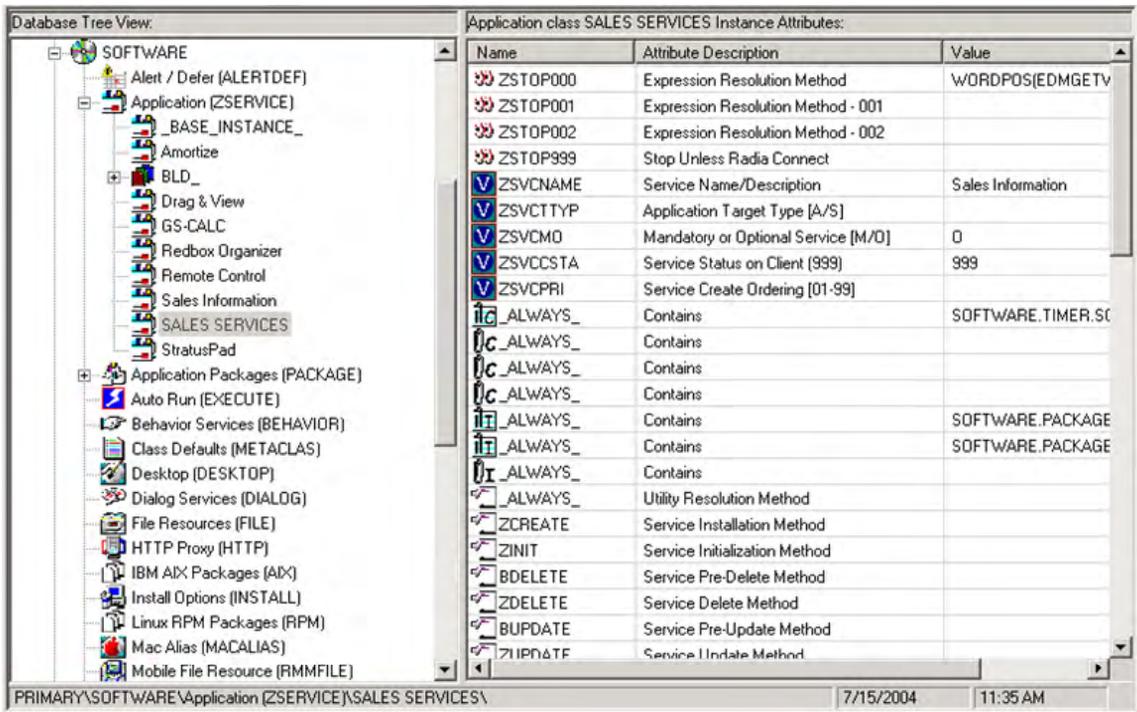


Now you can add services to the master service.

- 7 Right-click **SALES SERVICES**, and select **Show Connections** from the shortcut menu. The SOFTWARE.ZSERVICE Connections dialog box opens.

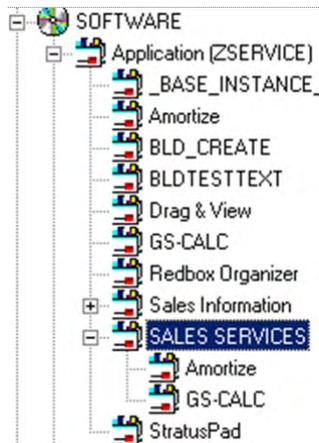


- 8 Select the **Application (ZSERVICE)** Class from the SOFTWARE.ZSERVICE connections dialog box, and click **OK**. The ZSERVICE Class instances are displayed on the list view of the screen.



- 9 Select the services from the right, and drag them to the ZSERVICE.SALES SERVICES instance.
- 10 Drop them onto the service when the cursor changes into a paper clip.
- 11 Confirm the connection.
- 12 Follow the steps that are outlined above to add services, if needed. The master service instance for this service group is now complete.

▶ The services grouped into a master service will install from top-down according to the order in which they are connected to the master service instance in the database. Member services in a master service group do not honor the settings in ZSVCPRI.



This master service is now ready to be deployed to your users.

Notify

Notify enables the Administrator to cause one or more agent computers to connect to the Configuration Server and to update or remove a specified application. It can also send e-mail notification to agent computers, based on the applications that Client Automation manages for them.

▶ For detailed information about Notify requirements, refer to the Deploying Applications chapter of the *Application Manager and Application Self-service Manager Guide*.

Notify communicates with agent computers that are members of an audience list. An agent computer is added to the audience list when Client Automation installs an application to that computer.

There are a number of prerequisites for using Notify:

- The subscriber must be connected to the Configuration Server before the notification. Notify is designed to notify only agents whose information is in the PROFILE File in the CSDB.
- The subscriber must have already installed the application using Client Automation in order to appear in the audience list for the specified application.
- If choosing to use email to notify subscribers, be sure that the email address for the subscriber is in the EMAIL variable of the USER Instance in the POLICY Domain, USER Class.
- The Configuration Server must be properly configured for Notify. For information on setting the appropriate values for the Configuration Server Settings file, `edmpprof`, refer to the Engineering Note, *OV-EN018720: Configuring the Radia Configuration Server for E-mail Notification* on the HP Software support web site.



To locate this document:

- Go to **www.hp.com/go/hpsoftwaresupport**.
 - Click **Use self-solve knowledge search**.
 - Enter your HP Passport User ID and Password.
 - Enter the Engineering Note number into the **Enter keyword(s), questions, phrases, or document identifiers** text box.
 - Select **Exact match**.
 - From the scrolling **Product** list, select **Client Automation management applications**.
 - Select **All Versions, All Operating Systems, and Technical documents**.
 - Click **Search**.
- Select how the CSDB Editor should generate the Notify command for agents from the Notify tab of the Options dialog box. You can customize the UID and STARTDIR values when notifying subscribers to install an application. For details, see the [Notify Tab](#) on page 136.

The CSDB Editor offers two ways to initiate a Notify action:

- Drag a POLICY Class instance (USER, DEPT, or WORKGRP) to an Application (ZSERVICE) instance. The audience list will consist of all agent computers that are members of the POLICY Class instance and for which Client Automation has installed and manages the application represented by the application instance.
- Select **Notify Subscribers** from the shortcut menu associated with an application instance. The audience list will consist of agent computers for which Client Automation has installed and manages the application represented by the application instance.



In either case, the application must already be installed on the agent computer before that computer is eligible to be notified.

After the Notify is initiated, CSDB Editor presents the same dialog boxes to complete the notification process, regardless of how the Notify was initiated.

Filtering—Viewing Portions of the CSDB

As your desired state grows, you will find it convenient to restrict the CSDB Editor to view only a portion of the CSDB. This saves effort in scrolling through the window to locate the domain, class, or instance you need to inspect or edit. The CSDB Editor has the ability to set filters to provide user-defined restricted views of the CSDB. This is particularly valuable for classes that can have thousands of instances such as the USER Class or the FILE Class.

The filter specification tells the CSDB Editor which domains, classes, or instances to display. Filtering is based upon matching the name of the domain, class, or instance with the filter specification you provide.

Filter specifications can contain one or more wildcards, represented by an asterisk, and/or one or more placeholder characters represented by a question mark. An asterisk matches any number of characters in the name of the component being filtered. Each question mark represents any one character in the name of the component being filtered.

Here are a few examples of valid filter expressions, and what they select.

Table 47 Examples of valid filter expressions

Filter Specification	Selects
*	Wildcard to select all; removes any filter currently in effect.
_*	Names whose first character is an underscore.
_A	Names that have a prefix, and any level of the prefix is followed by an A .
*.?LL	Names ending in a period, followed by any one character, followed by LL .
-??WORD.HP?	Names containing any number of characters preceding a dash, followed by any two characters, followed by WORD , followed by any number of characters up to a period, followed by HP , followed by any one character.

You can apply filters that remain in effect for an entire CSDB Editor session (*permanent filters*), or filters that apply only to the current CSDB Editor window (*temporary filters*).

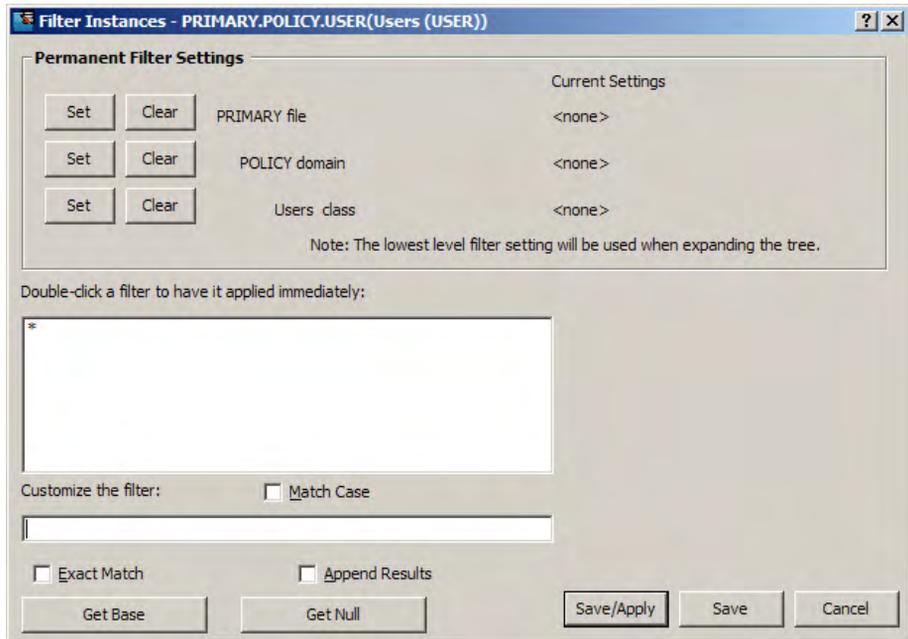
Permanent Filters

Permanent filters are applied every time you open any CSDB Editor window and remain in effect for that entire CSDB Editor session. Each class can have its own permanent filter specification.

To set or clear a permanent filter use the Permanent Filter Settings group box at the top of the Filter dialog box.

To set a permanent filter to filter instances of the **USER Class** on the prefix **ESB_**

- 1 Expand the **POLICY Domain** in the tree view.
- 2 Scroll down the tree view until you reach the **USER Class**.
- 3 Right-click the **USER Class**. A shortcut menu opens.
- 4 Click **Filter Instances** to open the Filter Instances dialog box.



- 5 Type **ESB_*** in the Customize the filter text box.
- 6 In the Permanent Filter Settings group box, click **Set** next to Users class. The filter specification is displayed.
- 7 To apply the filter, click **Save/Apply** at the bottom of the dialog box. The dialog box closes, and the tree display will be updated with the filter applied.

- 6 Click **Save/Apply**.

To remove a temporary filter

- 1 Apply * as a filter specification.

This matches any domain, class, or instance name, effectively removing the filter. The * filter specification is frequently found in the *express filter* list.

- 2 Click **Save/Apply** to remove the filter.

Viewing the Base or Null Instance

The Filter Instances dialog box also allows you to restrict your view to only the Base Instance or the Null Instance of the class.

To view the base or null instance

- 1 Expand the POLICY Domain in the tree view.
- 2 Scroll down the tree view until you reach the USER Class.
- 3 Right-click the **USER Class**.
- 4 Select **Filter Instances** from the shortcut menu. The Filter Instances dialog box opens.
- 5 Click **Get Base** to view the `_BASE_INSTANCE_` of the USER Class, or click **Get Null** to view the `_NULL_INSTANCE_` of the USER Class.
- 6 Click **Save/Apply**.

Appending Filter Results

You might want to continue viewing the current filter's result, and add an additional filter. To view both results at the same time, select the **Append Results** check box when you perform the next filter.

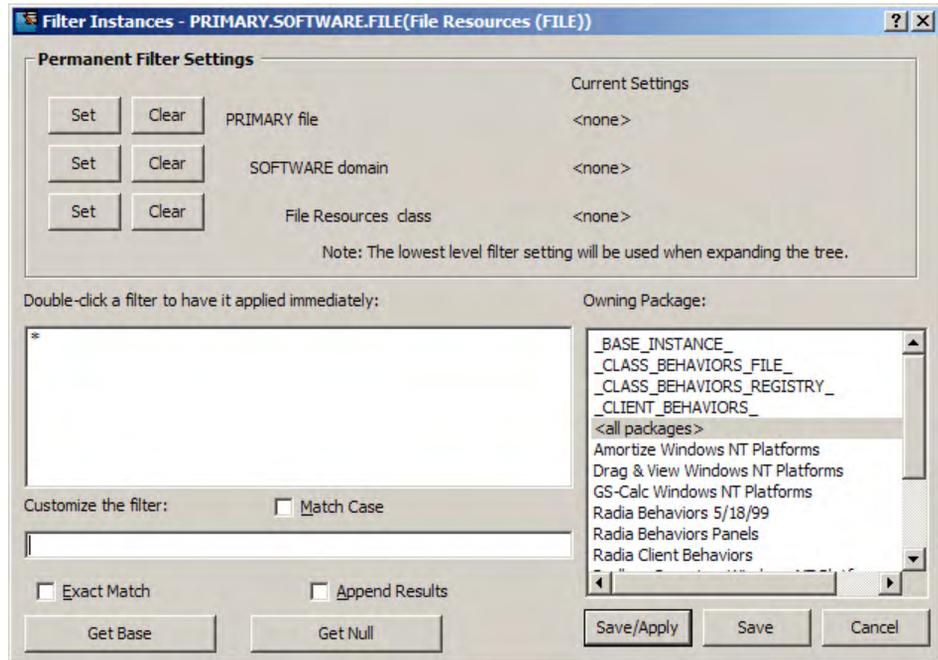
Filtering Component Class Instances

Component class instances (such as those of the FILE and DESKTOP Classes) can be filtered based on the configuration class instance to which they are connected.

When filtering component classes (such as the FILE class of the SOFTWARE Domain in this example), the Filter Instances dialog box contains a scrolling

list labeled Owning Package, as shown in [Figure 27](#) below. The package instances are listed in this scrolling list.

Figure 27 Filtering component classes



To filter FILE instances to select only those connected to a particular PACKAGE instance

- Double-click a package listed in the Owning Package list.
- or
- Click the package in the Owning Package list, and click **Save/Apply**.

For example, select **Amortize Windows NT Platforms** in the Owning Package list, and then click **Save/Apply**. The Filter Instances dialog box will close, and the selected instances will appear in the list view of the CSDB Editor window.

Express Filters

The box beneath the label Double-click a filter to have it applied immediately contains a list of potentially useful filter specifications that an administrator with appropriate authority can customize.

These are express filter specifications. As the label indicates, simply double-click the specification of your choice to apply it immediately. The filter is temporary, applying only to the current CSDB Editor window.

Customizing Express Filters

You can specify the lists of express filters that appear in the Filter Instances dialog box by storing them in the CSDB, and making any needed connections. The CSDB Editor retrieves the list of express filters for a particular Filter Instances dialog box from the CSDB as the dialog box opens.

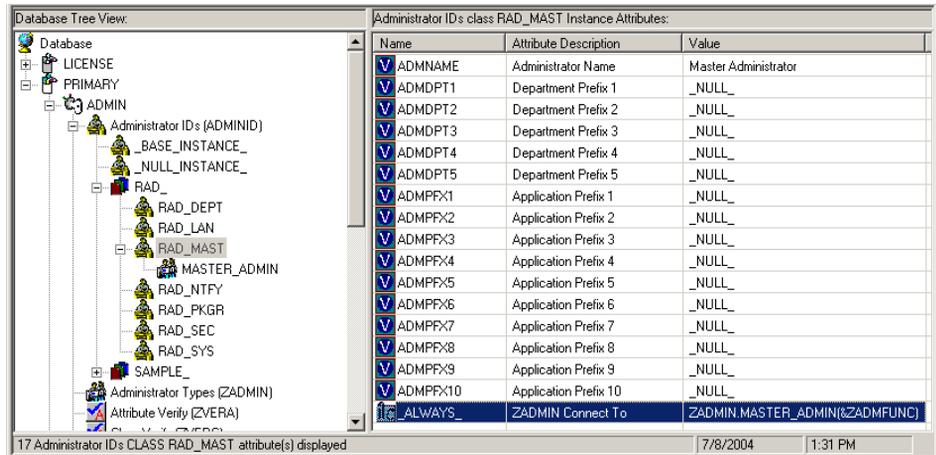
The CSDB Editor locates the correct list in the CSDB by starting a resolution process based on the user ID that was entered when the CSDB Editor session was started, and the Client Automation component (i.e., domain, class, or instance) being filtered. See Resolution Process in the *Essentials Guide* for more information.

The list of express filters for a particular domain is taken from an instance of the ZLIST32 class of the ADMIN Domain in the PRIMARY File. Here's how:

Assume you have logged on as user ID RAD_MAST. The authority and defaults associated with an administrator's user ID are stored in the ADMIN Domain of the PRIMARY File. The ADMINID Class contains an instance for each administrator's user ID (in this case, the instance name is RAD_MAST).

The RAD_MAST instance contains a connection to an instance of the ZADMIN class, in this case ZADMIN.MASTER_ADMIN.

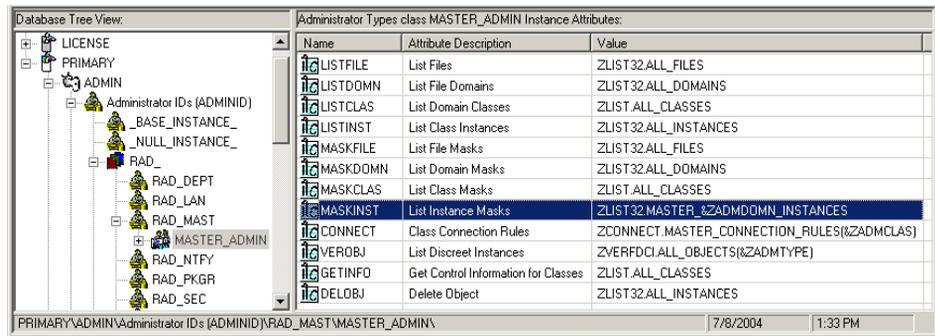
Figure 28 The RAD_MAST instance connections



The connection to ZADMIN.MASTER_ADMIN passes a parameter in the system message (&ZADMFUNC) that indicates which of the connections

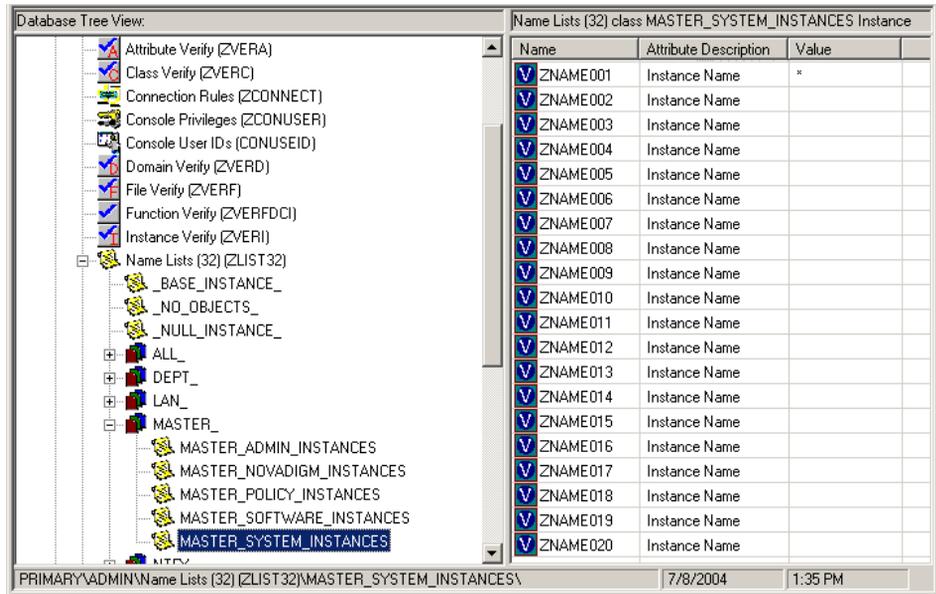
contained in the ZADMIN.MASTER_ADMIN instance should be made. Only those connections whose names match the value of the system message (or whose name is `_ALWAYS_`) are actually made during a particular resolution. The CSDB Editor sets the parameter's value before initiating the resolution. In this case, the CSDB Editor sets the value of the ZADMFUNC variable (and thus, after symbolic substitution, the value of the system message) to MASKINST, because the Filter Instances dialog box needs to retrieve a list of instance masks.

Figure 29 ZADMFUNC variable set to MASKINST



This causes a connection to an instance of the ZLIST32 class, ZLIST32.MASTER_ & ZADMDOMN_INSTANCES, based upon the domain to be filtered (SYSTEM). The domain is dynamically set in the ZADMDOMN variable by the CSDB Editor. This is determined by the class the administrator right-clicked to access the Filter Instances dialog box. In our example, the connection will be made to ZLIST32.MASTER_SYSTEM_INSTANCES (after the resolution process performs symbolic substitution), which contains the desired list of default express filter specifications.

Figure 30 Connection to LIST32.MASTER_SYSTEM_INSTANCES



To change the list of express filters for a domain, edit the appropriate instance of the ZLIST32 class. The changes will affect all administrators whose user IDs connect to the ZLIST32 instance you modify.

To set custom express filters for particular administrators, use the information detailed above to connect their user ID instances in the ADMIN.ADMINID Class to the appropriate ZLIST32 instance containing the desired list of express filters.

See [Adding an Instance](#) on page 156 and [Editing an Instance](#) on page 157 for information on how to use the CSDB Editor to make these changes.

Maintaining Connection Rules in the CSDB

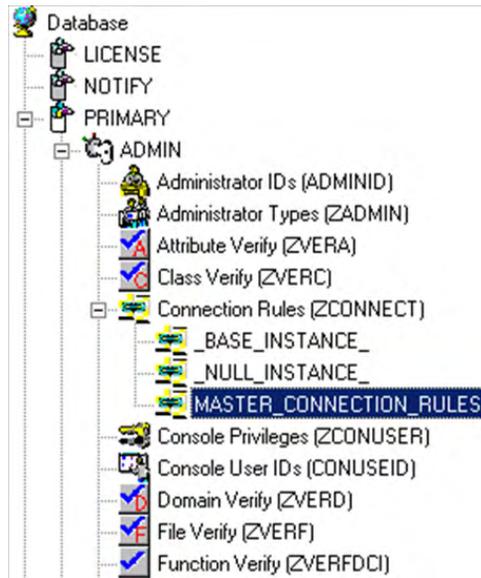
The CSDB Editor provides drag-and-drop connections between instances of classes. These connections, which contain connection rules, are stored in the CSDB.

Connection Rules Overview

To allow drag-and-drop capabilities for a new class, you need to establish rules for the new class. These connection rules are stored in the PRIMARY

File, ADMIN Domain, ZCONNECT Class, MASTER_CONNECTION_RULES Instance.

Figure 31 MASTER_CONNECTION_RULES instance



While the ZCONNECT Class contains the connection rules, the Name List (ZLIST) Class contains all of the rules for all of the classes. In order to create a **connection rule**, you need to create the rule in the ZLIST Class. Then you add the rule to the ZCONNECT Class.

Figure 32 Name Lists (ZLIST) Class in ADMIN Domain

The screenshot shows the CSDB Editor interface. On the left is the 'Database Tree View' showing the hierarchy of the ADMIN domain, with 'Name Lists (8) (ZLIST)' selected. On the right is a table titled 'Name Lists (8) (ZLIST) Class Instances:'.

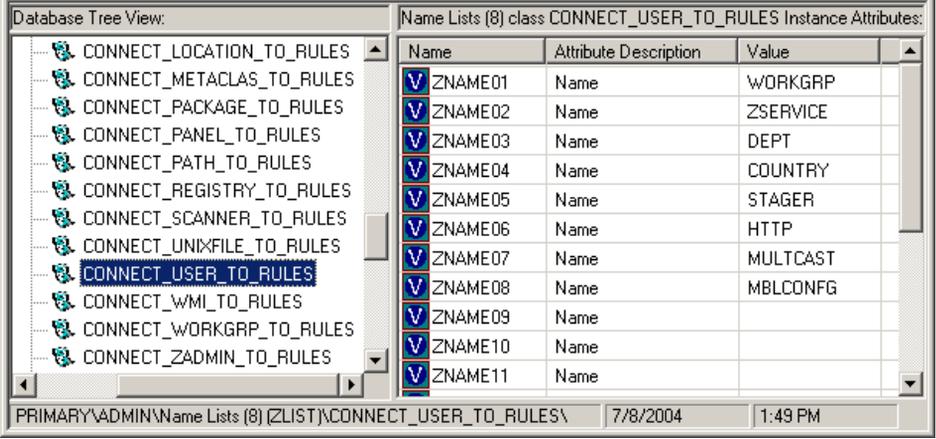
Name	Instance Name	Type
_BASE_INSTANCE_	_BASE_INSTANCE_	ADMIN.ZLIST Instance
_NULL_INSTANCE_	_NULL_INSTANCE_	ADMIN.ZLIST Instance
ALL_CLASSES	ALL_CLASSES	ADMIN.ZLIST Instance
CONNECT_ADMINID_TO_RULES	CONNECT_ADMINID_TO_RULES	ADMIN.ZLIST Instance
CONNECT_BEHAVIOR_TO_RULES	CONNECT_BEHAVIOR_TO_RULES	ADMIN.ZLIST Instance
CONNECT_CONUSEID_TO_RULES	CONNECT_CONUSEID_TO_RULES	ADMIN.ZLIST Instance
CONNECT_DEPT_TO_RULES	CONNECT_DEPT_TO_RULES	ADMIN.ZLIST Instance
CONNECT_DESKTOP_TO_RULES	CONNECT_DESKTOP_TO_RULES	ADMIN.ZLIST Instance
CONNECT_DIALOG_TO_RULES	CONNECT_DIALOG_TO_RULES	ADMIN.ZLIST Instance
CONNECT_FILE_TO_RULES	CONNECT_FILE_TO_RULES	ADMIN.ZLIST Instance
CONNECT_INSTALL_TO_RULES	CONNECT_INSTALL_TO_RULES	ADMIN.ZLIST Instance
CONNECT_LOCATION_TO_RULES	CONNECT_LOCATION_TO_RULES	ADMIN.ZLIST Instance
CONNECT_METACLAS_TO_RULES	CONNECT_METACLAS_TO_RULES	ADMIN.ZLIST Instance
CONNECT_PACKAGE_TO_RULES	CONNECT_PACKAGE_TO_RULES	ADMIN.ZLIST Instance
CONNECT_PANEL_TO_RULES	CONNECT_PANEL_TO_RULES	ADMIN.ZLIST Instance
CONNECT_PATH_TO_RULES	CONNECT_PATH_TO_RULES	ADMIN.ZLIST Instance
CONNECT_REGISTRY_TO_RULES	CONNECT_REGISTRY_TO_RULES	ADMIN.ZLIST Instance
CONNECT_SCANNER_TO_RULES	CONNECT_SCANNER_TO_RULES	ADMIN.ZLIST Instance
CONNECT_UNIXFILE_TO_RULES	CONNECT_UNIXFILE_TO_RULES	ADMIN.ZLIST Instance
CONNECT_USER_TO_RULES	CONNECT_USER_TO_RULES	ADMIN.ZLIST Instance
CONNECT_WM1_TO_RULES	CONNECT_WM1_TO_RULES	ADMIN.ZLIST Instance
CONNECT_WORKGRP_TO_RULES	CONNECT_WORKGRP_TO_RULES	ADMIN.ZLIST Instance
CONNECT_ZADMIN_TO_RULES	CONNECT_ZADMIN_TO_RULES	ADMIN.ZLIST Instance
CONNECT_ZCONNECT_TO_RULES	CONNECT_ZCONNECT_TO_RULES	ADMIN.ZLIST Instance
CONNECT_ZSERVICE_TO_RULES	CONNECT_ZSERVICE_TO_RULES	ADMIN.ZLIST Instance
CONNECT_ZVERC TO RULES	CONNECT_ZVERC TO RULES	ADMIN.ZLIST Instance

At the bottom of the window, the path 'PRIMARY\ADMIN\Name Lists (8) (ZLIST)' is shown, along with the date '7/8/2004' and time '1:47 PM'.

The CSDB Editor then uses these rules to populate the class list in the Show Connections dialog box, and to control drag-and-drop connections.

For example, let's look at the connection rules that have been established for the USER Class. To do this, examine the ZLIST.CONNECT_USER_TO_RULES Instance as shown in [Figure 33](#) on page 205.

Figure 33 ZLIST.CONNECT_USER_TO_RULES Instance



Name	Attribute Description	Value
ZNAME01	Name	WORKGRP
ZNAME02	Name	ZSERVICE
ZNAME03	Name	DEPT
ZNAME04	Name	COUNTRY
ZNAME05	Name	STAGER
ZNAME06	Name	HTTP
ZNAME07	Name	MULTICAST
ZNAME08	Name	MBLCONFG
ZNAME09	Name	
ZNAME10	Name	
ZNAME11	Name	

An instance of this particular USER Class can contain a connection to any of the following classes:

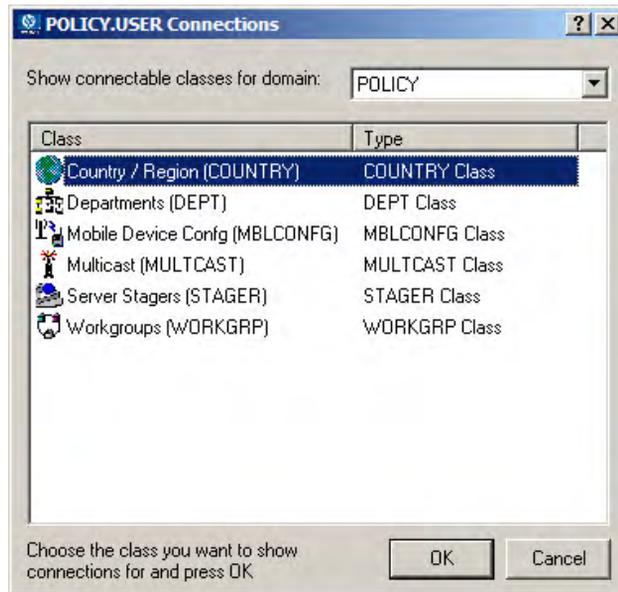
- WORKGRP
- ZSERVICE
- DEPT
- COUNTRY
- STAGER
- HTTP
- MULTICAST
- MBLCONFG

The ZLIST instance contains a list of up to 20 classes to which the class, represented by the ZLIST instance (in this case USER), can connect. You can edit this list to include additional classes, or remove classes from a connection rule. If any class in the CSDB needs to be able to connect to more than 20 different classes, you can edit the ZLIST Class template and add more ZNAME nn variables.

After you establish the rules for connection in the Name List (ZLIST) Class, the connection type can be created. Since we are creating a connection rule for drag-and-drop, the class variable is added to the Connection Rules (ZCONNECT) Class. Then edit the Rules variable to define the type of connection that can be made to the USER Class.

Now you can make a drag-and-drop connection to the new class. [Figure 34](#) below shows the connectable classes for the POLICY Domain where the USER Class is located.

Figure 34 Connectable classes for the POLICY Domain



Enabling Connections in New Classes

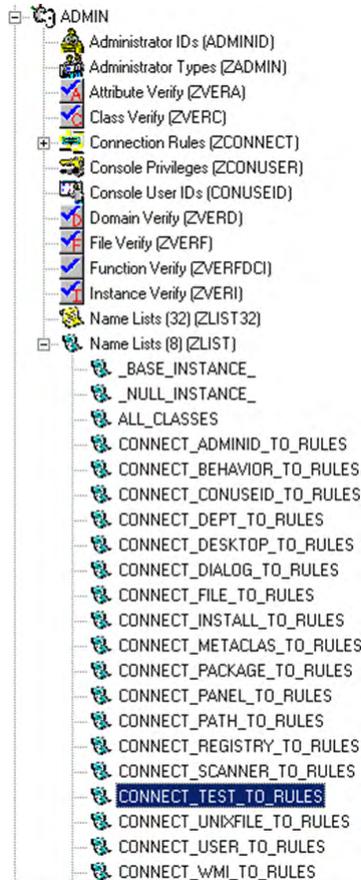
If you add a new class to the CSDB that can validly contain connections to other classes, you need to:

- Create an instance in the ZLIST Class to represent the new class.
- Set the new ZLIST Class instance to allow the connection rules.
- Add the new class to the ZCONNECT Class template.
- Set the new class' variable in the MASTER_CONNECTION_RULES.

For example, the steps below show how to add the connections for a new class called TEST.

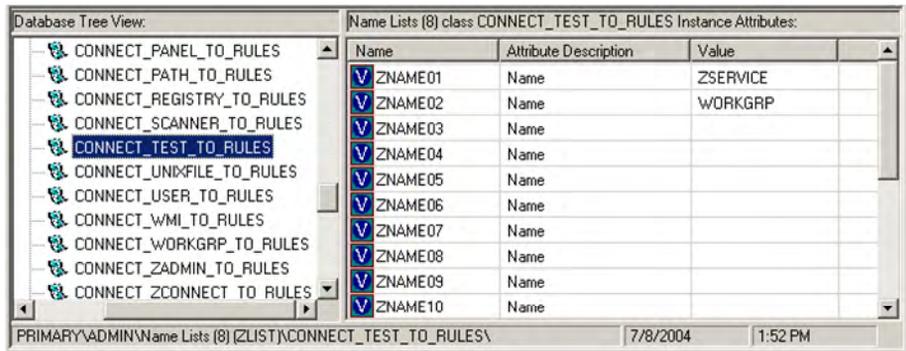
- 1 Create an instance of the ZLIST Class named `CONNECT_newclass_TO_RULES`, where *newclass* is the name of the new class, for example, TEST.

- 2 Open the CSDB Editor and navigate to the PRIMARY.ADMIN.Name Lists (8) (ZLIST) Class.
- 2 Right-click **ZLIST** and select **New Instance**.
- 3 Name the new instance as suggested above, **CONNECT_TEST_TO RULES**. This figure shows the added **CONNECT_TEST_TO_RULES** in the ZLIST Class.



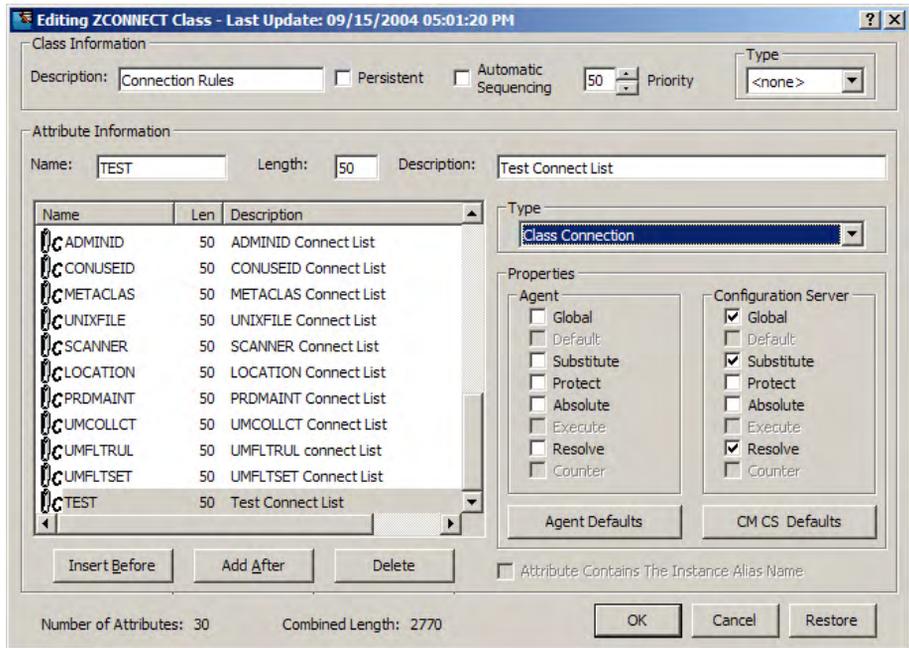
- 4 Edit the **CONNECT_TEST_TO_RULES** Instance, setting each **ZNAME n** variable to the name of a class to which the new class can be connected.

For example, set values for **ZNAME01** and **ZNAME02** to allow connections to the **ZSERVICE** and **WORKGRP** Classes, as shown in the following figure.



Now you need to add the **connection rule** variable to the Connection Rules (ZCONNECT) Class.

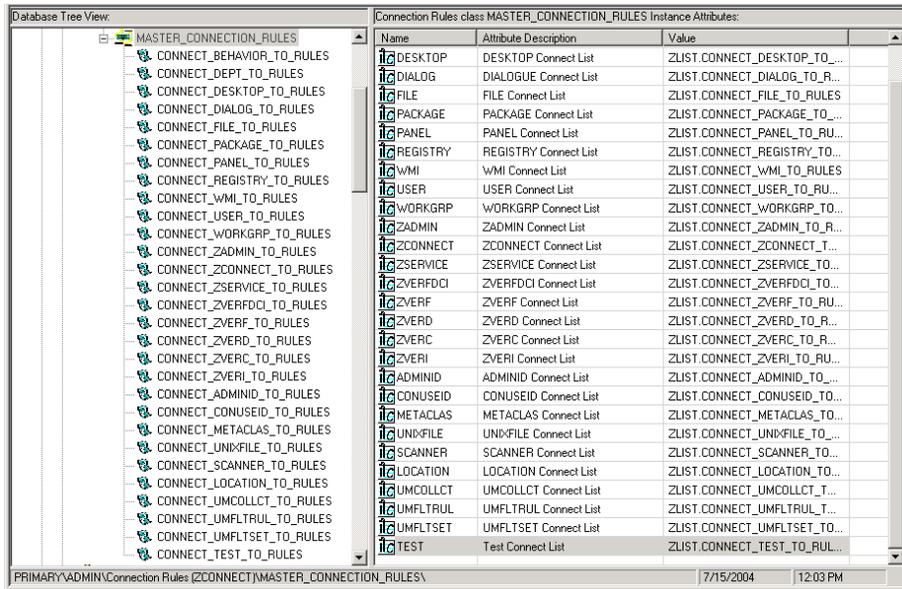
- 5 Navigate to the PRIMARY.ADMIN.ZCONNECT Class.
- 6 Right-click on the **ZCONNECT** Class and select **Edit Class** from the shortcut menu.
- 7 Add the new class variable (TEST) to the class template for the ZCONNECT Class. The length of the attribute is 50, and the description is Test Connect List.
 - When adding an attribute to the ZCONNECT Class, be sure to change the attribute type to Class Connection. See the figure below.
 - For information on editing a class, see [Editing a Class](#) on page 145.



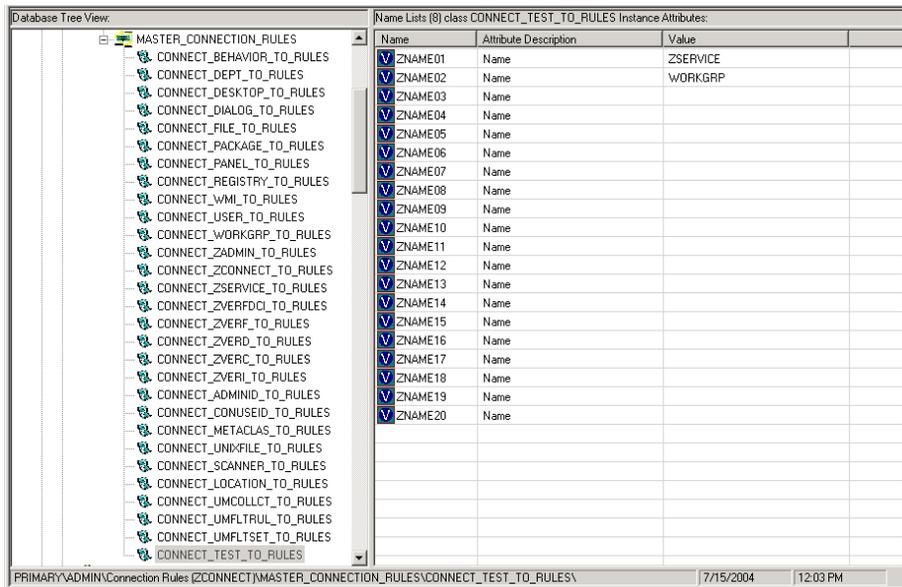
After you add the new attribute to the class template, you need to set the value of the new attribute in the ZCONNECT.MASTER_CONNECTION_RULES Instance.

- Locate the new instance in the list view (right side of the screen), and set the value of the new variable to ZLIST.CONNECT_TEST_TO_RULES, where *newclass* is the name of the new class.

For example, check the value that has been set for the TEST attribute in the next figure.



The next figure shows that the new connection rule was added to the MASTER_CONNECTION_RULES, thereby allowing drag-and-drop connections for the new class.



Managing WI-enabled Applications

This section describes the Feature Set Editor.

Feature Set Editor Overview

The CSDB Editor has a Feature Set Editor, with a wizard type interface, that guides you through the selection and configuration of the features of a Windows Installer-enabled software package managed by Client Automation. This allows you to use the features of Windows Installer while taking advantage of the packaging and deployment technologies of Client Automation.

With the release of Office 2000, Microsoft introduced new technology (Microsoft Windows Installer) for installing and maintaining software products on the Windows platform. Client Automation products can install and manage Windows Installer-enabled applications as follows:

The Publisher allows for *basic* and *advanced* methods of publishing and configuring Windows Installer-enabled applications.

 For information regarding the packaging of Windows Installer-enabled applications, see the *Extensions for Windows Installer Getting Started Guide*.

For information regarding the publishing of Windows Installer-enabled applications, see [Publishing Windows Installer Files](#) on page 88.

The CSDB enables HP products to store Windows Installer feature sets.

Client Automation provides policy administration and state data gathering.

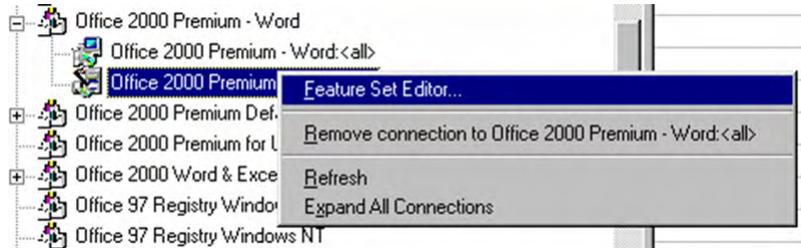
Client Automation provides the ability to control the customization of MSI packages without the need to create Windows Installer Transform (MST) files.

Client Automation provides the ability to discover and manage previously installed Windows Installer-enabled applications that were not deployed using CM.

Accessing the Feature Set Editor

To access the Feature Set Editor

- 1 Locate the Application Packages instance that represents the software package whose features you want to edit.
- 2 Double-click this instance to display its MSI and MSIFEATS connections.
- 3 Right-click the MSIFEATS connection in the tree view, as follows:



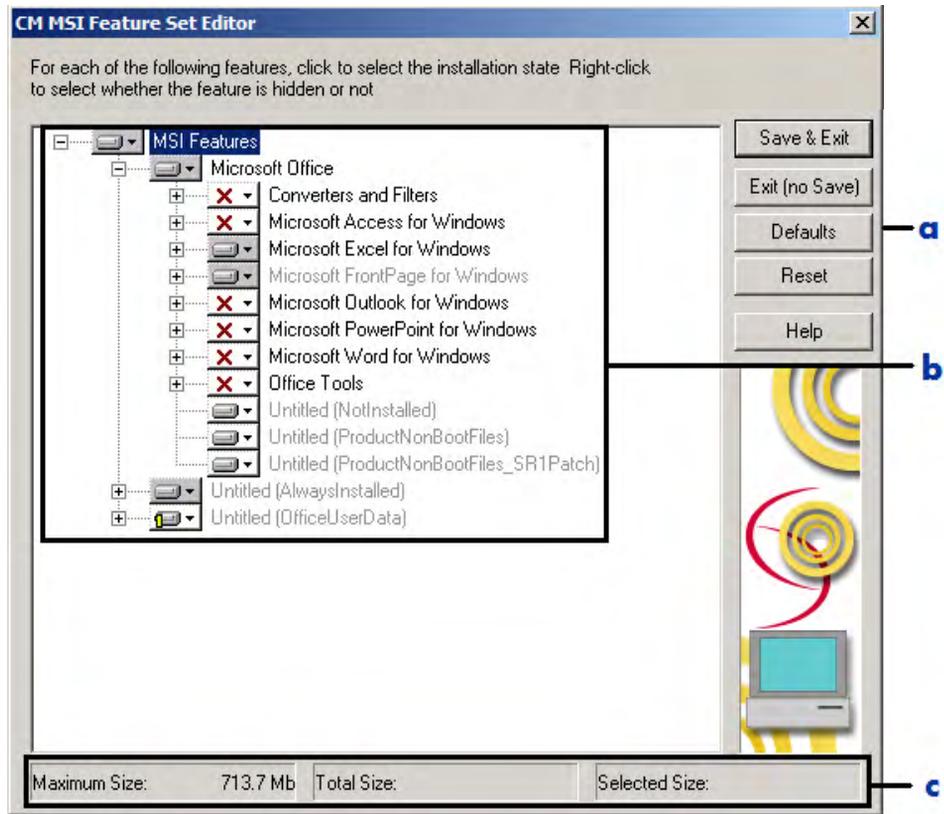
- 4 Select **Feature Set Editor** from the shortcut menu.

The Feature Set Editor dialog box opens.

About the MSI Feature Set Editor

Figure 35 on page 213 shows that the Feature Set Editor interface is divided into three areas, the **tree view** on the left, **buttons** on the right, and **package size information** on the bottom.

Figure 35 Feature Set Editor dialog box



Legend

- a Buttons
- b Tree view
- c Package size information

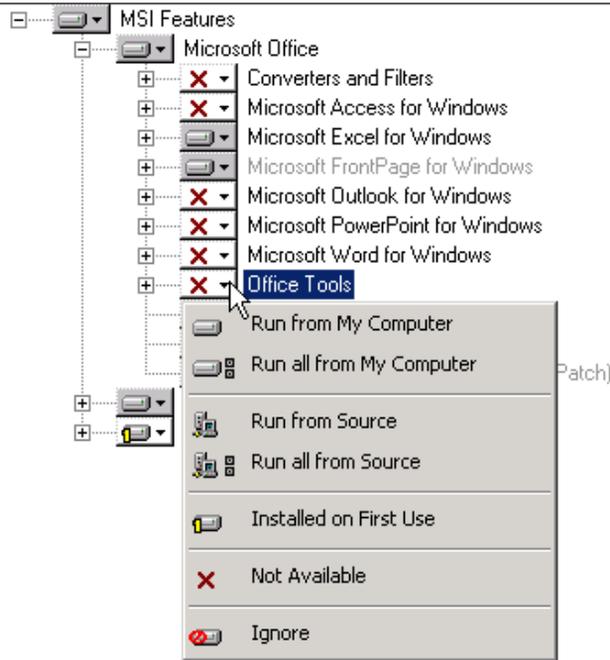
Tree View

Features are grouped into a hierarchy through which you navigate using the tree control. Higher-level features consist of two or more lower-level, child features. Navigate through the tree as you would through a typical Windows Explorer tree.

Features Menu

When you click on the drop-down button next to a feature, the Features menu opens.

Figure 36 Features menu



Each of the features has an attribute that indicates how that feature is to be installed and maintained.

Names of features whose attributes have been changed during the current session of the MSI Feature Set Editor appear in boldface in the tree.

Normally, by setting an attribute for a higher-level feature, the setting is automatically applied to all of the child features that comprise it. See the attribute descriptions in [Table 48](#) on page 215 for exceptions to this rule.

Table 48 Features menu options

Feature Icon and Name	Description of Function
 Run from My Computer	<p>Features are physically installed on a hard drive on the agent computer, and therefore perform best.</p> <p>This applies to all child features, unless the child feature's attribute is already set to Installed on First Use or Not Available.</p> <p>The children appear on branches of the tree, and are beneath and connected to the selected feature.</p>
 Run all from My Computer	<p>Features are physically installed on a hard drive on the agent computer. All features that are children have their attributes changed to Run from My Computer, and they, too, are physically installed on the agent computer.</p> <p>Features perform best because they are run from the end user's machine. Mobile computer users should have all of the features they need while traveling set to Run from My Computer or Run all from My Computer. Otherwise, when they use a feature that is not installed on their computer's hard drive, they will be prompted for the original installation media, which might not be available.</p>
 Run from Source	<p>Features are not physically installed on a hard disk on the agent computer; rather they are run from the Administrative Installation Point (AIP). This is normally a network share persistently accessible to the agent computer.</p> <p>When you select this attribute for a feature, the attribute is automatically applied to all features that are children, unless the child feature's attribute is already set to Not Available.</p> <p>The Run from Source and Run all from Source attributes save local hard drive space at the expense of slower performance and the need to have an active connection to the AIP.</p>
 Run all from Source	<p>Features are not physically installed on a hard disk on the agent computer, rather they are run from the AIP. This is normally a network share persistently accessible to the agent computer.</p> <p>When you select this attribute, the Run from Source attribute is applied automatically to all features that are children, regardless of the attribute currently assigned to the child feature.</p> <p>The Run from Source and Run all from Source attributes save local hard drive space at the expense of slower performance and the need to have an active connection to the AIP.</p>

Feature Icon and Name	Description of Function
 Installed on First Use	<p>Features are not installed on the agent computer until the end user tries to use the feature through a shortcut or menu selection. Shortcuts for shortcut-accessible features are deployed when the package is initially installed. When the end user first invokes the feature, files that are needed to support the feature are installed on the end user's hard disk, where they remain. Future uses of the feature are run from the agent computer.</p> <p>In order to install such a feature, the agent computer must be running the Redirector service, and it must be able to connect to the Configuration Server computer.</p> <p>This minimizes the use of hard disk space on agent computers by allocating space only to features that they actually use. On the other hand, end users are subject to the delay needed to install the feature at the time when they first invoke it.</p> <p>When you select this attribute, the Installed on First Use attribute is applied automatically to all features that are children, unless the child feature's attribute is already set to Not Available.</p>
 Not Available	<p>Features are not installed, are unavailable to the end user, but could affect other services. The attributes for all child features are also set to Not Available.</p> <p>For example, one service is defined to deploy MS Word, and all other features of Office are set to Not Available. Another service is defined to deploy MS Excel, and all other features of Office are set to Not Available.</p> <p>Now, imagine that the end user installs the MS Word service. It installs successfully and does not install any of the other features of Office. A bit later, the end user installs the Microsoft Excel service. The MS Excel service indicates that all other features of Office are Not Available. Therefore, MS Excel is installed, but MS Word is uninstalled.</p> <p>If the end user selects an option that requires the feature, the software instructs the end user to run the setup again and change the installation state of the feature. However, end users whose software is managed by Client Automation generally do not have access to the setup.</p> <p>Assign the Not Available attribute only to features that must be unavailable to end users. Provide information to the end user about what to do if they receive a message.</p>

Feature Icon and Name	Description of Function
 Ignore	<p>Note: This is available for backward compatibility. You do not need to determine which features need to be ignored by a service.</p> <p>Features set to Ignore are not installed, are unavailable to the end user, but will not affect other services. Client Automation management products act as though the feature were not a component of the managed software at all. The attributes for all child features are also set to Ignore.</p> <p>For example, one service is defined to deploy only MS Word, with all other features of Office set to Ignore. Another service is defined to deploy MS Excel and PowerPoint, setting all other features of Office to Ignore. If an end user installs both services, the features installed by the first-installed service will not be affected by the installation of the second-installed service.</p> <p>Both services can be created from one package published to the CSDB. The package is cloned by copying it with another name. You can then use the Feature Set Editor to edit the features of each package.</p>

Names of features whose attributes have been changed during the current session of the MSI Feature Set Editor appear in boldface in the tree.

Hide and Disable Features

When you right-click the name of a feature (in the following example, Microsoft Word for Windows), the shortcut menu for Hide and Disable opens:

Figure 37 Shortcut menu for Hide and Disable



When you select **Hide**, the feature and all of its child features will not appear in the native Microsoft Windows Installer Update Features dialog box, which

is accessible to the end user via the Control Panel's Add/Remove Programs applet.

All features that are hidden appear in the MSI Feature Set Editor tree with the prefix (H) before the name of the feature.

When you select **Disable**, the feature and all of its child features are ignored by the Client Automation agent. A disabled feature's value is determined by Windows Installer, from the combination of MSI and MST files, instead of the value in the Client Automation instance.

All features that are disabled appear in the MSI Feature Set Editor tree with the prefix [D] before the name of the feature

Feature Set Editor Buttons

The Feature Set Editor buttons are located on the right side of the Feature Set Editor window. [Table 49](#) below describes these buttons.

Table 49 Feature Set Editor buttons

Button	Description of Function
	Click Save & Exit to accept the changes that were made to the attributes of the file and exit the Feature Set Editor. It is also possible to exit the Feature Set Editor by clicking the X on the top right corner.
	Click Exit (no Save) to discard any changes that were made, and to then exit the Feature Set Editor. It is also possible to exit the Feature Set Editor by clicking on the X on the top right corner.
	Click Defaults to reset all features' attributes to the settings in the MSI file distributed with the software by the software vendor. The default setting will not be applied to the feature set until the feature set is saved when you click Save & Exit .
	Click Reset to discard any changes made to feature settings since you opened the current session of the MSI Feature Set Editor. Settings for all features are refreshed by re-reading them from the CSDB.
	Click Help to access the CSDB Editor Help topics.

Package Information

Use the package information to view the size of the current Windows Installer-enabled features that you are configuring.

Maximum Size

This is the total number of bytes that would be needed on the agent computer to install the features in the tree, *if all* features in the tree they were given the Run from My Computer or Run All from My Computer attribute.

Total Size

This is the total number of bytes needed on the agent computer to install only those features in the tree that have the Run from My Computer or Run All from My Computer attribute.

Selected Size

This is the total number of bytes needed on the agent computer to install a single feature (and its child features) that you have selected. The size calculation includes only features with the Run from My Computer or Run All from My Computer attribute.

Windows Add or Remove Programs Settings

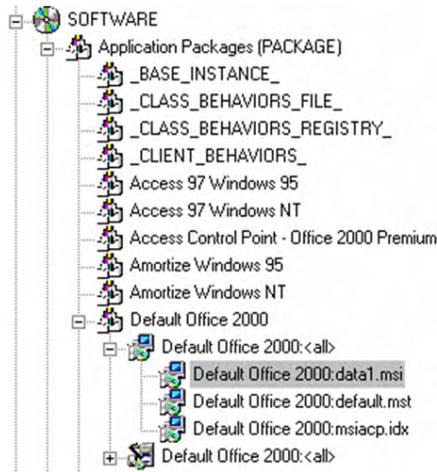
When a Windows Installer-enabled software package, such as MS Office, is installed on an agent computer, it is normally added to the list of applications that can be managed by the Add or Remove Programs application of Windows Control Panel.

Client Automation allows you to not list an application in the Add or Remove Programs application of Windows Control Panel. This prevents the end user from interfering with Client Automation products' centralized management of the application. Alternatively, Client Automation also allows you to selectively hide individual features from access by the Add or Remove Programs application.

Preventing an Application's Listing in Add or Remove Programs

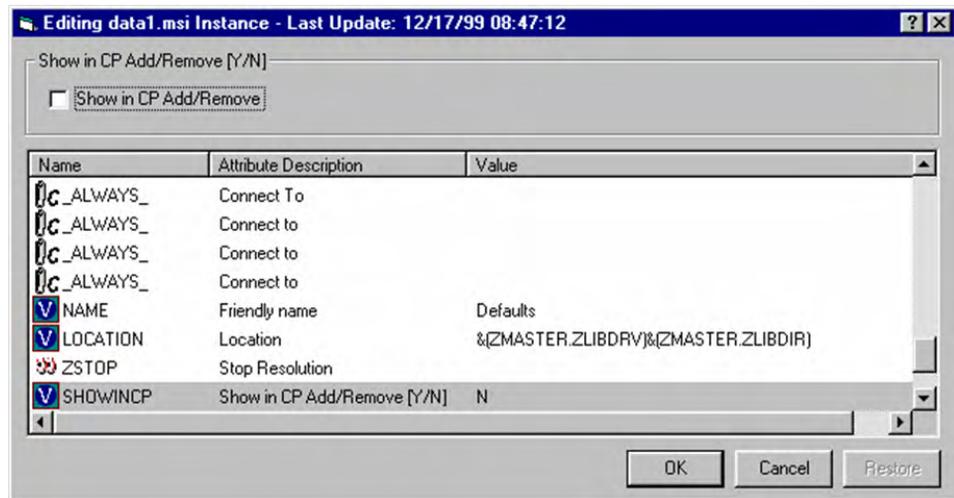
To prevent listing the application in the Add or Remove Programs application, use CSDB Editor to change the setting of the SHOWINCP variable of the MSI class instance for the `DATA1.MSI` file that is connected to the Application Packages Class instance:

Figure 38 DATA1.MSI instance



Change the value of the SHOWINCP variable to N, as follows:

Figure 39 Editing DATA1.MSI instance



Hiding Features displayed by Microsoft Windows Installer Update Features

When the end user selects the Windows Installer-enabled software package in the Add or Remove Programs application and clicks **Change**, Microsoft Windows Installer is invoked, and opens its native user interface for manipulating features of the installation on the agent computer. If the end

user clicks **Add or Remove Features**, Microsoft Windows Installer displays its Update Features windows.

Use the MSI Feature Set Editor's Hide function to prevent individual features from appearing in the Microsoft Windows Installer Update Features dialog box. This allows you to use Client Automation to control the end user's installation of the application software's features via your Client Automation management product's capabilities, and avoids possible end-user interference.

Copying a Windows Installer Enabled Package

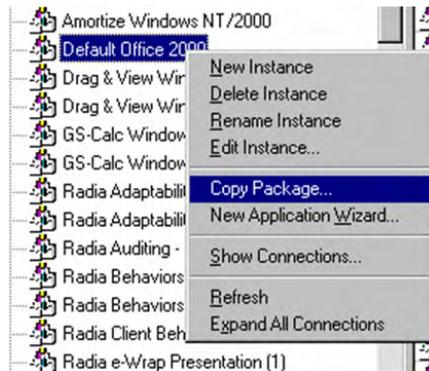
One of the key advantages of using Client Automation products to manage Windows Installer-enabled applications is the ability to specify variants of the software package, and make those variants available to targeted users.

When you use the advanced capabilities of Client Automation, you to create only one comprehensive AIP, and publish it to the CSDB. All variant sets of features are encapsulated in individual packages in the CSDB.

To make a new variant set of features

Copy an existing package, and edit it to meet its unique requirements.

- 1 Right-click on the Application Packages instance and select **Copy Package** from the shortcut menu.



A message tells you that copying a package instance will result in a duplication of all the package's component instances, and that this may take a while.

- 2 Select **Yes** to continue the Copy Package function.

Select **No** to cancel the Copy Package function.

Copying the package copies the package instance and all of the component class instances that belong to the package instance. The new package is a complete copy of the original. This operation maintains the integrity of the CSDB, because each package retains control over its own components.

- 3 In the Copy Package and Components dialog box, type the name for the newer copied package.
- 4 Click **OK** to continue. The Configuration Server takes a few moments to copy the package.
- 5 When the newly copied package appears in the CSDB Editor's tree view, right-click on the MSIFEATS instance beneath the package and select **Feature Set Editor** from the shortcut menu.

The MSI Feature Set Editor opens.

- 6 Set the properties of the applications that should not be deployed to Ignore.
- 7 Click **Save & Exit**, and the feature set for the package is updated. You can now connect this package to the appropriate policy (USER, DEPT, WORKGRP) instances to deploy and manage a single feature as a stand-alone application.

Managing Feature Sets in Native Mode

In some enterprises, packaging applications and administering Client Automation are separate functions. You might not want to use the MSI Feature Set Editor to edit the transform (MST) files. Use the following procedure to publish the package and MST files and allow you to manage feature sets in native MSI mode.

- 1 Use the Publisher in Advanced mode to create the ACP and publish the Windows Installer package.

After you publish the package, use CSDB Editor to remove the connection to `SOFTWARE.MSIFEATS.packageprefix_*` from the Application Packages (PACKAGE) instance.

- 2 If you have multiple MST files, use the Publisher in Component Select Mode to promote each of the MST files into individual packages. This will create a new PACKAGE instance for each MST file.
- 3 Connect the MSI PACKAGE instance to each of the transform file's PACKAGE instances.

- 4 Create one Application (ZSERVICE) instance for each of the transform packages.
- 5 Assign subscribers to the appropriate transform package's service.

 Any changes to or replacement of the *.MST file will cause Client Automation to call MSIEXEC to uninstall and reinstall the application.

When the service is installed or modified, Client Automation will detect the absence of the MSIFEATS object, and will use the native MSI mode, MSIEXEC. Client Automation will call MSIEXEC natively with the proper location of the data source. If Client Automation calls MSIEXEC and detects the presence of a single MST file, it will add the .MST file to the command line that invokes MSIEXEC. If Client Automation calls MSIEXEC and detects more than one MST file, it will run MSIEXEC only for the MSI file. The MSI native behavior will be used for update and repair.

 If you are using an MST file to include a new feature that is not already a feature available in the Feature Set Editor (MSIFEATS), the new feature from the MST file will be included in the MSI service. However, if there is a feature in the MST file that already exists in the MSIFEATS object, the feature state indicated in MSIFEATS will be used, and the one in the additional MST file will be disregarded. If you want to manage the MSI package using only your MST file, use the procedure described in this section.

The Registry File Editor

Use the **Registry File Editor** to edit the properties or values for registry keys associated with a Client Automation package.

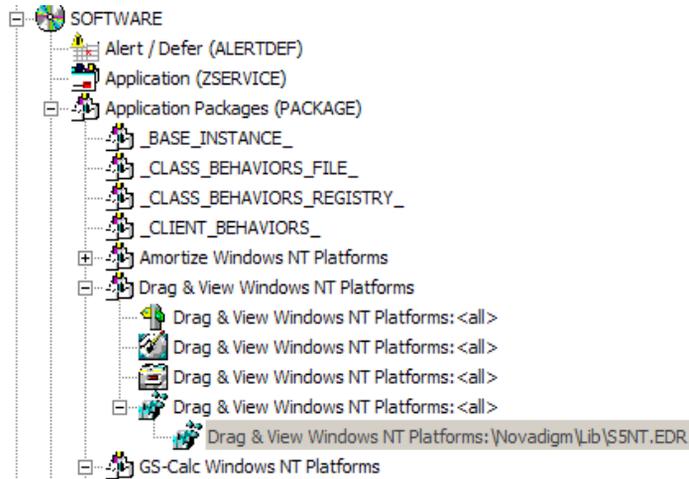
When you use Packager to create a package, all of the registry changes for that package are stored in a file (with an .EDR extension) in the LIB directory. Use the Registry File Editor to modify this file. Then, the file is re-published to the CSDB and the instance is updated.

Accessing the Registry File Editor

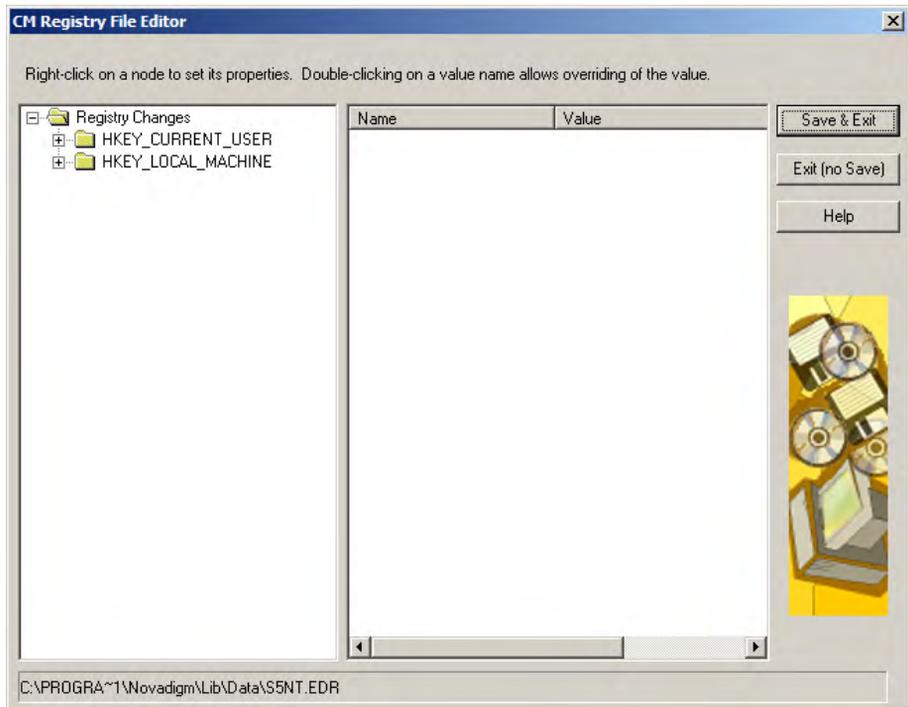
To access the Registry File Editor

- 1 Locate the Registry Resources instance that represents the software package whose registry settings you want to change.

- 2 Right-click the instance in the tree view. A shortcut menu opens. For example, locate the registry instance for the Drag & View Windows NT Platforms package and right-click it.



- 3 Select **Edit Registry Resource**. The Registry File Editor dialog box opens.



- The tree view (on the left) displays nodes that represent keys in the registry file.
- The list view or value name (in the middle) displays the contents of a selected registry key.
- The buttons (on the right) allow you to Save & Exit, Exit without saving, or access the Help function in the Registry File Editor.

Set Properties Shortcut Menu

To set properties for the Registry File Editor, right-click any node of the tree view in the Registry File Editor. A shortcut menu opens.

Each option in this shortcut menu is described in the following sections.

Active

Select **Active** to include the registry keys in the package.

Inactive

Select **Inactive** to prevent the registry keys from being included in the package. Inactive nodes appear with a  symbol covering its folder.

Deploy/Verify

Click **Deploy/Verify** to access the following submenu:

- **Unconditional/Enforced**
Client Automation deploys the registry key every time the subscriber connects to the Configuration Server, even if it already exists on his desktop. This setting repairs the registry key if it is modified or deleted. Use this setting for registry keys that must remain invariant in order for the application software to function properly.
- **Unconditional/Exist**
Client Automation deploys the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset if the user modifies it. This is the default setting.
- **Exist/Exist**
Client Automation deploys the registry key only if it does not exist on the subscriber's desktop, during both the initial deployment and on each subsequent connection. This gives users full control over the registry key value, but reinstalls the key if it is accidentally deleted.

New Key

Click **New Key** if you would like to add a new value to your registry. This opens a dialog box to enter the New Key name. Click **OK** to accept the new key, or **Cancel** to return to the Registry File Editor.

New Value

Click **New Value** to add a value to a newly created key or to add a value to an existing registry key. This opens a dialog box where you enter the following information:

Value Name

Enter the name of the value you want to add to the registry file. This name will appear in the Value Name (center) of the Registry File Editor window.

Data Type

Use the drop-down menu to select from the list of available data types. The type selected is as follows:

- **REG_SZ**
A text sequence most typically used for descriptive information. This is the default setting.
- **REG_DWORD**
Data represented by an integer that is 4 bytes long and is displayed in the binary, hexadecimal, or decimal format.
- **REG_EXPAND_SZ**
An expandable data string. This text contains a variable that is replaced when called by an application.
- **REG_MULTI_SZ**
A string containing multiple text entries of user-readable text.

Delete Always

Click **Delete Always** if you want all values under a Registry key to be deleted during a Radreg32 Delete operation. When this option is turned on, all values under the key are deleted *without* restoring their original registry values in the backup EDR file.

Delete Never

Click **Delete Never** if you want to retain all values under a key during a Radreg32 Delete operation.

Delete (Default)

Click **Delete (Default)** if you want the Radreg32 Delete operation to exhibit normal Delete behavior with regards to deleting or restoring values.

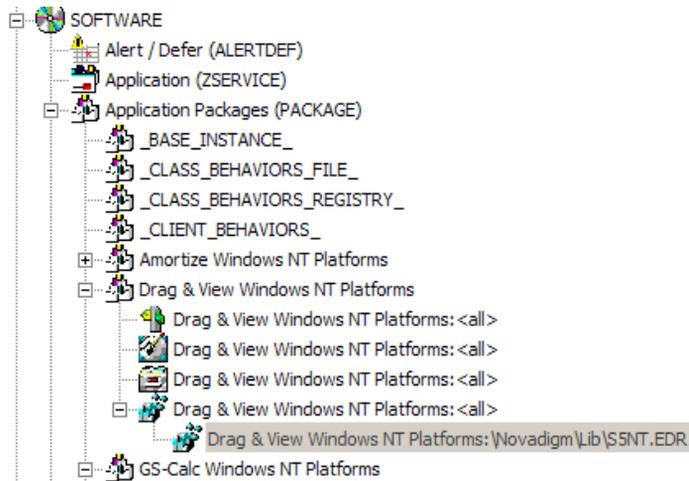
Editing a Registry Key

To edit a registry key's value access the key through the Registry File Editor. Follow the example in this section to edit the information in a registry key.

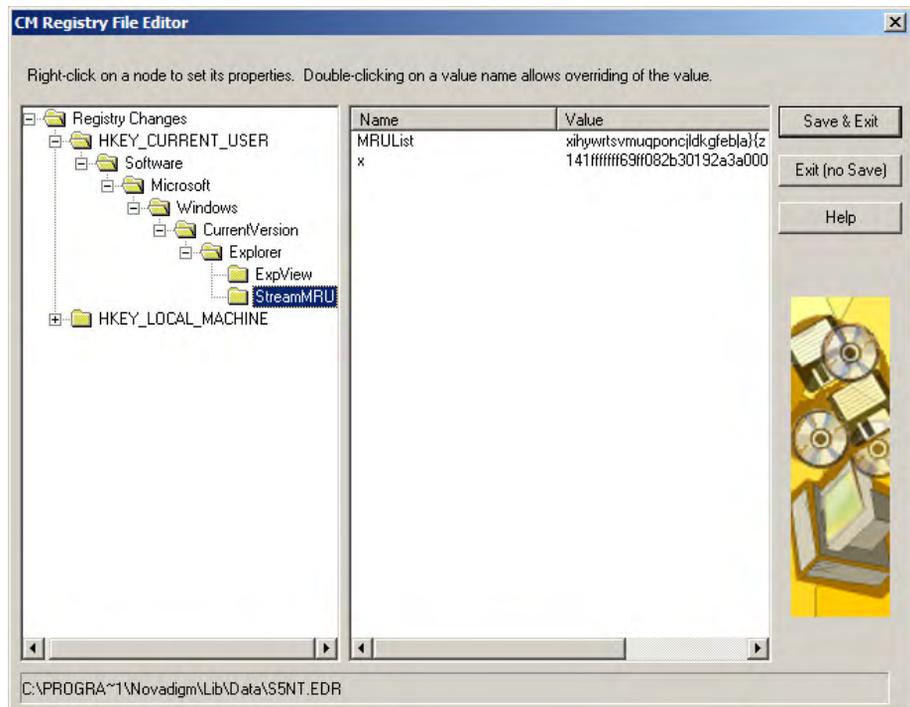
- ▶ Use extreme caution when you use the Registry File Editor. Editing registry keys without thorough understanding of their functions could render a package or service useless.

To edit the data in a registry key

- 1 From the Windows taskbar, click **Start → Programs → HP Client Automation Administrator → Client Automation Administrator CSDB Editor**. The CSDB Editor Security Information dialog box appears.
 - ▶ The factory set user ID is RAD_MAST. No password is necessary. Your Client Automation security administrator might have changed this during installation. Check with him or her to obtain your own user ID and password, if necessary.
- 2 If necessary, type a user ID and password, then click **OK**. The CSDB Editor splash screen appears briefly.
- 3 Double-click **PRIMARY**.
- 4 Expand the SOFTWARE Domain and then the Application Packages (PACKAGE) Class.
- 5 Expand the package Drag & View Windows NT Platforms.
- 6 Expand the Registry instance.



- 7 Right-click the registry file and select **Edit Registry Resource** from the shortcut menu. The Registry File Editor opens.
- 8 Click the node to expand and display the contents of the registry key in the list view.
- 9 Double-click the name of the registry key in the list view.



The Override Value dialog box opens.

- 10 Edit the data in the New Value text box.
- 11 Click **OK** to save the changes and close the dialog box.

or

Click **Cancel** to close the dialog box without saving the changes.

To restore the data in a registry key

- 1 Double-click the name of the registry key in the list view.
- 2 Click **Reset**.
- 3 Click **OK** to save the changes and close the dialog box.

or

Click **Cancel** to close the dialog box without saving the changes.

Closing the Registry File Editor

Close the Registry File Editor when you are finished making changes to instances of the Registry file.

To close the Registry File Editor

- 1 Click **Save & Exit** to save the changes and close the dialog box.

or

Click **Exit (no Save)** to discard the changes and close the dialog box.

- 2 Click **Yes** to confirm that you want to save the changes and close the dialog box.

or

Click **No** to return to the Registry File Editor.

AMP Editor

The AMP Editor is a tool you use to create and modify Application Profiles. The *HP Client Automation Application Management Profiles User Guide (AMPs Guide)* describes Application Management Profiles and how to edit them.

4 Administrator for Linux and Macintosh

At the end of this chapter you will:

- Understand the system requirements and permissions necessary to install the Administrator.
- Be able to install the Administrator using either the graphical or non-graphical mode.
- Understand how to use the Agent Explorer for Linux.
- Understand how to use the Publisher.

Introduction

The Administrator for LINUX has two components: Publisher and Agent Explorer.

The Administrator for Macintosh has one component: Publisher.

Installation

This section describes how to install the Administrator for LINUX and Macintosh.

System Requirements

- TCP/IP connection to a computer running the Configuration Server.

Platform Support

For detailed information about supported platforms, see the release note document that accompanies this release.

Prerequisites

- We strongly recommend installing the Client Automation agents as root (or as a Mac OS X administrator user ID capable of issuing **sudo** commands).
- Install the Client Automation agent on a local file system.
- If you plan to run any of the graphical components of the Administrator software, make sure the LINUX environment variable DISPLAY is set in your environment. If it is not, the installation will default to a non-graphical installation. However, you can set this variable to indicate the hostname or IP address to which you would like to redirect the graphical display.

Table 50 **Setting the DISPLAY Variable**

In a....	Type....
C shell	setenv DISPLAY IP address or hostname:0.0
Bourne, Bash, or Korn shell	DISPLAY=IP address or hostname:0.0 export DISPLAY

- ▶ If the DISPLAY environment variable is not set in your environment, the installation defaults to a non-graphical installation.

Troubleshooting

If you encounter any problems while installing the Client Automation agent, perform the following steps before contacting technical support:

- Enable diagnostic tracing by appending the text `-loglevel 9` to the installation command line and re-run the installation.
 - ▶ The install option `-loglevel 9` should only be used to diagnose installation problems.
- Access `tmp/setup/setup.log` located in the home directory of the user ID who ran the install.

Recommendations

- We strongly recommend that you install and run the Administrator as root.
 - ▶ Root authority is required to apply owner and group designators to managed resources.

Installation Methods

To install the Administrator do either of the following:

- Execute the installation procedure directly from the installation media.

- Copy the files from the installation media into a temporary directory and execute the installation procedure.

You can use several parameters on the command line when you install the Administrator.

`-mode plain`

Example: `./install -mode plain`

Installs the Administrator in plain mode. The installation graphics are displayed in plain mode (no moving graphics). This is useful for remote installations where network bandwidth might be an issue.

`-mode text`

Example: `./install -mode text`

Installs the Administrator in text mode using the non-graphical installation. The installation takes place completely on the command line. The installation will default to text mode if the `DISPLAY` environment variable is not set.

Installing the Administrator for LINUX or Macintosh

This section describes how to install the Administrator for LINUX or Macintosh.

Graphical Installation

This section describes how to install the Administrator for LINUX or Macintosh using a graphical user interface (GUI).

To install the Administrator for LINUX or Macintosh using a GUI



If the user ID of the person performing the Administrator installation has previously installed a Client Automation agent, the location of the Administrator will default to the location of the Client Automation agent executables.

- 1 Depending on your operating system, change your current working directory to the correct subdirectory on the installation media.

Example: For RedHat:

```
/Configuration Server/management_infrastructure/  
administrator_workstation/linux/
```

- 2 Type `./install` (for Mac OS X, type `sudo ./install`) and then press **Enter**. The Welcome window opens.
- 3 Click **Next**. The HP Software License terms window opens.
- 4 Read the software license terms and click **Accept**.



If you are installing the Administrator to a computer with the same user ID that had previously installed a Client Automation agent, the installation program will prompt you for the Configuration Server's IP address next. The next three windows: Administrator Location, Lib directory, and Log directory, are needed only if you are installing the Administrator to a computer that does not have a Client Automation agent already installed.

The Administrator Location window opens.

- 5 Type the name of the directory where you are installing the Administrator, or click **Browse** to navigate to it.
- 6 Click **Next**. If the specified directory already exists you will be prompted to verify this location.
 - If you would like to update the existing directory, click **OK**.
 - If you would like to change the directory location, click **Cancel**.

The Lib Directory window opens.

- 7 Type the name of the directory where you would like to store proprietary information created by Client Automation (the `lib` directory), or click **Browse** to navigate to it.
- 8 Click **Next**. The Log Directory window opens.
- 9 Type the name of the directory where you would like to store the log files generated by Client Automation, or click **Browse** to navigate to it.
- 10 Click **Next**. The Configuration Server IP Address window opens
- 11 Type the IP address (format: `xxx.xxx.xxx.xxx`) or host name of the Configuration Server you will be publishing to.
- 12 Click **Next**. The Configuration Server Port Number window opens.
- 13 Type the port number of your Configuration Server (default is 3464).
- 14 Click **Next**. The Package Settings window opens.

- 15 Review the settings displayed in the Package Settings window. If you would like to change any of the settings, click **Back** until you get to the appropriate window.
- 16 When you are satisfied with the Package Settings, click **Install**. The Administrator is installed.

Command-Line Installation

This section describes a non-graphical (using a command line) installation of the Administrator for Linux or Macintosh.

To install the Administrator for Linux or Macintosh using a command line

- 1 At a command prompt, change your current working directory to the correct operating system subdirectory on the installation media.

Example: For HP-UX:

```
/Configuration Server/management_infrastructure  
/administrator_workstation/hpux/
```

- 2 Type `./install -mode text` (for Mac OS X, type `sudo ./install -mode text`) and then press **Enter**.

The Administrator installation begins.

- 3 Type **C**, and then press **Enter**.



If you are installing the Administrator to a computer with the same user ID that had previously installed a Client Automation agent, the installation program will prompt you for the Configuration Server's IP address next. The next three sections: Administrator Location, `Lib` directory, and `Log` directory, are only needed if you are installing the Administrator to a computer that does not have a Client Automation agent already installed.

- 4 Specify the installation location for the Administrator, and then press **Enter**.
- 5 Specify the location for the Client Automation proprietary objects (`IDMLIB`), and then press **Enter**.
- 6 Specify the location for the log files created by Client Automation (`IDMLOG`), and then press **Enter**.
- 7 Specify the IP address of the Configuration Server, and then press **Enter**.

- 8 Specify the port number of the Configuration Server, and then press **Enter**.
- 9 Review the installation settings you have chosen. If you want to install the Administrator with these settings, press **Enter** to accept the default (Y) and begin the installation or type **N**, to re-enter your installation information.
- 10 To complete the configured installation process, press **Enter**. The Administrator is installed.

Agent Explorer

The Agent Explorer is available for Linux, but is not available for Macintosh.

Opening the Agent Explorer

The following instructions explain how to open and close the Agent Explorer.

To open the Agent Explorer

- From the **HP Client Automation Administrator** folder, click **Client Automation Administrator Agent Explorer**.

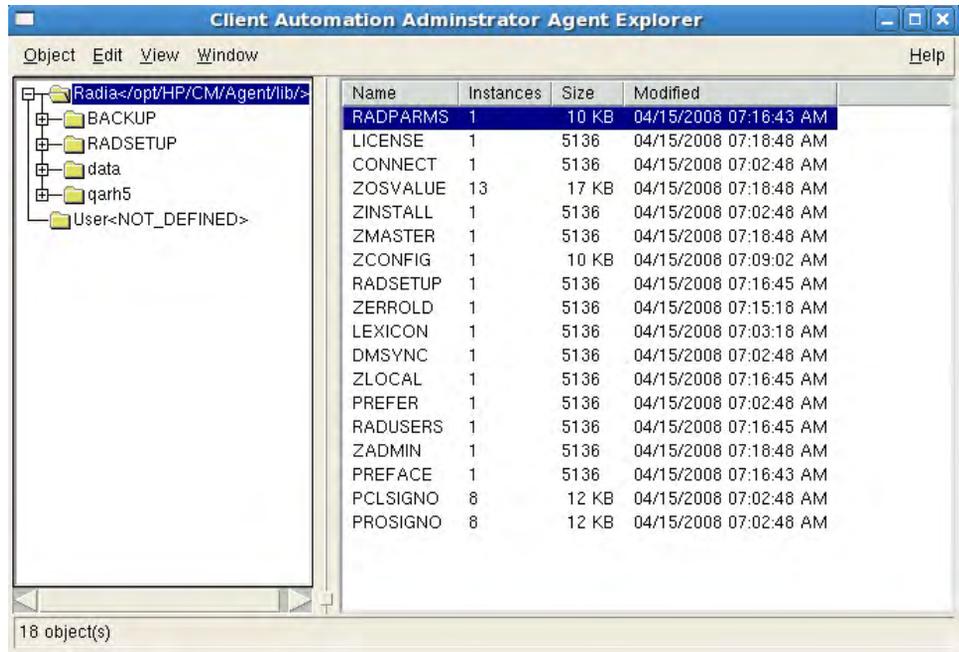
or

at the command line type `./radobjed`.

The Agent Explorer opens with the object list displayed in the right pane. See [Figure 40](#) on page 238. The object list contains the names of Client Automation objects that you can view and edit.

- ▶ Objects that begin with "Z" are used by Client Automation. When naming an object that you create, we recommend that you do not click a name beginning with the letter "Z."

Figure 40 Agent Explorer window



Menus in the Object List Window

Use the menus in the Agent Explorer window to open, copy, delete, rename, and create new objects, and to navigate the Agent Explorer.

Object Menu

Use the object menu to open, copy, delete, rename, and create new objects.

Figure 41 Object menu in the object list window



Table 51 Object menu options

Menu option	Function
New	Click New and type a name for the new object in the Create New Object dialog box.
Open	Select an object in the object list and click Open . The object view window opens where you can edit and manage variables and heaps for that object.
Copy	Select an object in the object list. Click Copy , and type a name for the new object. A new object is created with the same variables and values.
Rename	Select an object in the object list. Click Rename . Type a new name for the object.
Delete	Select an object from the object list. Click Delete to remove it from the object list. After it is deleted, an object, its heaps, and its variables cannot be recovered.
Change User Directory	
Exit	Click Exit to close the Agent Explorer.

Edit Menu

Select All

Click **Select All** to select all the objects in the object list. You can then perform actions from the Object menu on all objects at the same time.

View Menu

Refresh

Refreshes the screen.

Table Window Menu

Close all

Click **Close all** to close all open objects.

Menus in the Object View Window

The object view window displays the contents of a Client Automation object for you to view or edit, and contains four drop-down menus: Object, Edit, View, and Window.

Object Menu

Use the object menu to view information about the object, save changes you make, and close the object.

Figure 42 Object menu

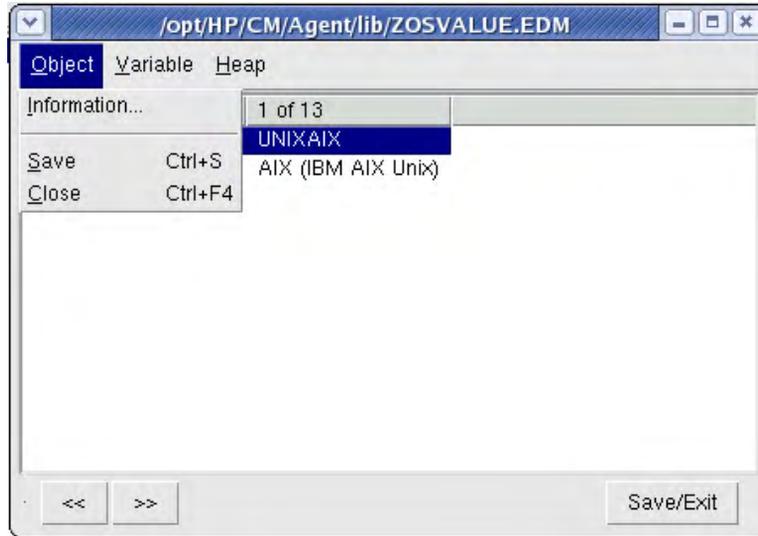


Table 52 Object menu options

Menu option	Function
Information	Displays the Object Information screen.
Save	Saves the changes you made.
Close	Closes the object view window.

Variable Menu

Use the variable menu to add, delete, and rename variables, and manage and navigate through the heaps in an object based on their variable's values.

Figure 43 Variable menu

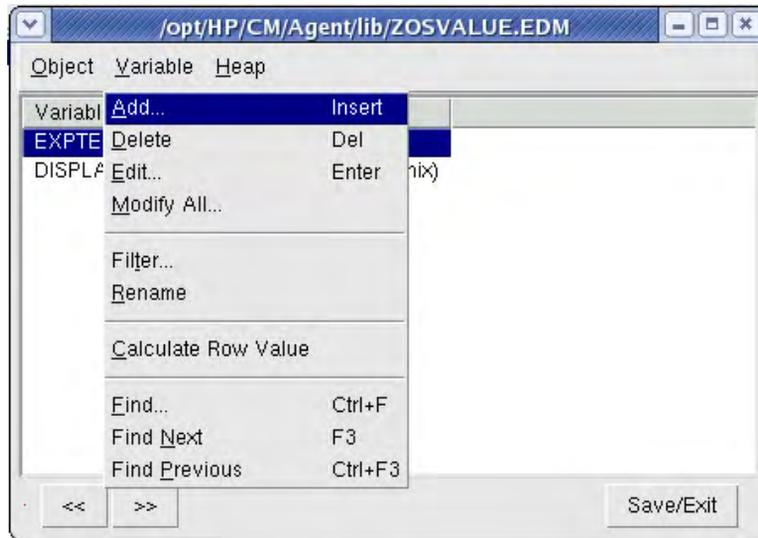


Table 53 Variable menu options

Menu option	Function
Add	Adds a variable with a user-specified name to an object.
Delete	Deletes a variable from an object.
Edit	Changes the value of a variable.
Modify All	Changes the value of this variable for all heaps in an object.
Filter	Displays all the variables that contain a user-specified character string.
Rename	Renames a variable in an object.
Calculate Row Value	For a selected variable, sums all the decimal only entries for all heaps in the object.
Find	For a user-selected variable, finds the heaps whose values contain a user-specified character string. Available only for multi-heap objects.

Menu option	Function
Find Next	After you use the Find command to enter a character string to find, this command finds the next occurrence of this character string.
Find Prev	After you use the Find command to enter a character string to find, this command finds the previous occurrence of this character string.

Heap Menu

Use the Heap menu to add, delete, and copy instances in an object, and to navigate through the instances in an object.

An instance is also called a heap. To view each instance in an object, click >> or << at the bottom of the Object View window.

Figure 44 Heap menu

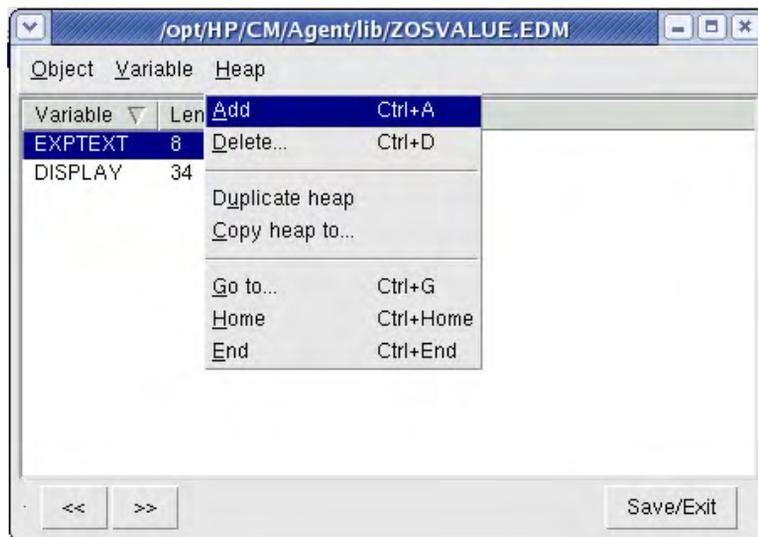


Table 54 Heap menu options

Menu option	Function
Add	Adds an instance with blank values. Go to Variable → Edit to enter the values.

Menu option	Function
Delete	Deletes the instance you select. Click >> or << to navigate to the instance that you want to delete.
Duplicate heap	Makes a copy of the instance you select, including its values. Click >> or << to navigate to the instance you want to duplicate.
Copy heap to	Copies the instance that you select to the object that you type in the Copy Instance to dialog box.
Go to	Jumps to the instance whose number you type in the New Instance # dialog box.
Home	Jumps to the first instance in an object.
End	Jumps to the last instance in an object.

Packager

The Packager is not available for Linux or Macintosh. For both of these operating systems, the package feature is part of the Publisher.

Publisher

The Publisher is available for the Linux and Macintosh operating systems. This section describes how to use both.

The Publisher for Linux and Macintosh only has one publishing mode – Component Select.

In Component Select mode, you select the individual components that make up the application, such as files, directories, and links to create a package.

Publishing in Component Select mode involves four phases:

- 1 **Select** the individual files to be published.
- 2 **Edit** the file properties and methods.
- 3 **Configure** the package and service options.

- 4 **Publish** the files to the CSDB.

Prerequisites

Before you publish the application in Component Select mode, install the target application on your packaging machine. This ensures that the files you need to select reside on the computer.

Publishing

This section describes the typical publishing steps of Component Select mode, and by referring you to the Publisher for Windows, provides detailed information about each screen that you encounter. Note that publishing in the Publisher for Linux and Macintosh are identical to publishing for Windows except for the delivery options. For Linux and Macintosh, the User Component and Machine Component options are not available. See Task 2 below.

To publish in Component Select Mode

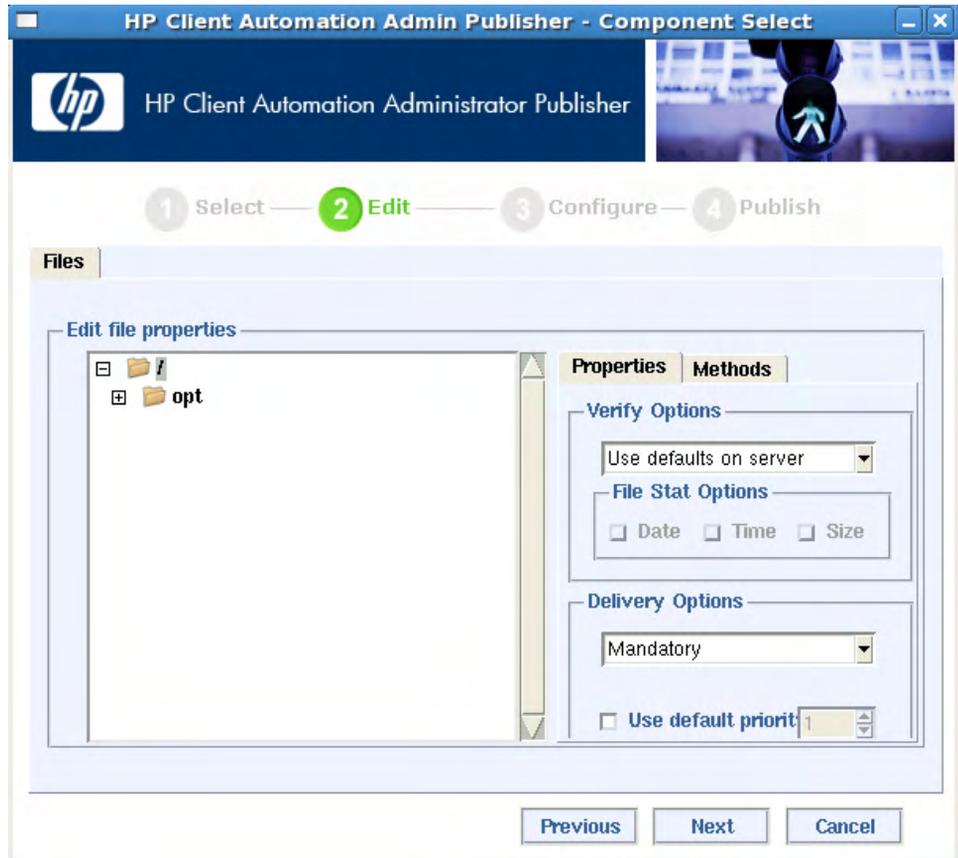
- 1 Log on as root.
- 2 To launch the Publisher run `./publishr` from the location where you installed the Administrator.
- 3 Type your User ID and Password in the appropriate fields.
 -  The User ID, as shipped, is `RAD_MAST`. No password is necessary. This may have been changed in your installation. Check with your Client Automation security administrator to obtain your own User ID and Password, if necessary. The User ID and password are case-sensitive.
- 4 In the Type of data to publish drop-down box, leave the default choice of **Component Select**. (This is the only publishing mode currently available for UNIX and Macintosh.)
- 5 Click **OK**. The Select window opens.

Task 1 Select Files to Publish

Use the Select files to publish window to select all files that you want to include in the package. This process is the same as it is for the Publisher for Windows. See [Select Files to Publish](#) on page 74.

Task 2 Edit File Properties and Methods

Use the Edit window to edit file and folder properties and methods. This process is the same as it is for the Publisher for Windows, except for Delivery Options. For the Linux and Macintosh version of the Publisher, the User Component and Machine Component options are not available.



See [Edit Properties and Methods](#) on page 76 for the similarities.

Task 3 Enter Package and Service Information

Use the Configure window to name the package and include additional descriptive information as well as set package deployment limitations based on hardware and operating system settings. Use the Service information window to enter descriptive information about the service, and information about how it will be managed after it is deployed. These processes are the same as they are for the Publisher for Windows, except for the options in the

Class drop-down list. For Linux the only class available is UNIXFILE, and for Macintosh, the only class available is MACFILE.

See [Enter Package and Service Information](#) on page 85 for the similarities.

Task 4 Publish the Package

Use the Publish window to view the package and service information and then to publish it. This process is the same as it is for the Publisher for Windows. See [Publish the Package](#) on page 87.

UNIX File Resources (UNIXFILE)

During the publication process, the UNIXFILE attributes are defined. These attributes define the owner and group associations and permissions of each published resource. Each package published has a corresponding UNIXFILE instance in the PRIMARY File. Use the CSDB Editor to view and modify these attributes.

▶ The following instructions use the CSDB Editor. The CSDB Editor is currently available for 32-bit Windows platforms. For more information, see [CSDB Editor](#) on page 125.

To view the UNIXFILE Class instances using the CSDB Editor

- 1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator CSDB Editor**.

The CSDB Editor Security Information dialog box opens.

- 2 If necessary, type a User ID and Password, and then click **OK**.

▶ The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the New Password and Verify New Password text boxes.

- 3 Double-click **PRIMARY**.
- 4 Double-click **SOFTWARE**.
- 5 Double-click **Unix File Resources (UNIXFILE)**.

- 6 Double-click the appropriate application. The attributes for the UNIXFILE instances for that application appear in the list view.

To change any instance attribute, double-click the attribute name in the list view. Make changes in the box that opens, and click **OK**.

Published Owner, Group, and Permission Considerations

The UNIXFILE Class contains the attributes ZPERUID and ZPERGID. They define the user ID and group association of the promoted resource. These attributes are populated during the publishing session and reflect the user ID and group association of the resources being promoted. In addition, permission characteristics are captured during publishing and stored in the UNIXFILE.ZRSCRASH attribute. These attributes can be changed using the CSDB Editor.

Table 55 Attributes exclusive to the UNIXFILE Class

Attribute	Description
ZPERUID	User ID associated with the promoted resource. The resource will be owned by this user ID when deployed, providing the HPCA Application Manager is run by root and the user ID exists on the agent workstation.
ZPERGID	Group ID associated with the promoted resource. The resource will be associated with this group when deployed, providing the HPCA Application Manager is run by root and the group exists on the agent workstation.
ZRSCRASH	This should be a four-digit octal notation of the managed resources permissions (example: 7555). This is populated during the publishing session based on the characteristics of the published resources.
DIRPERMS	Permissions assigned to unmanaged folders that are created when the directory structure does not exist. Client Automation assigns permissions based on umask settings if this attribute is not defined. For example, if the managed file <code>file.txt</code> is placed in <code>/opt/newlocation</code> , and the directory <code>/newlocation</code> does not exist, it will be created. This is an unmanaged directory. If more than one file is installed to a new location, the first one installed determines the assigned permissions.

If the Client Automation Agent is run as a non-root user ID:

- All deployed resources will be associated with the user ID and group of the user ID who is running the HPCA Application Manager.
- During publishing, the owner and group of the resource is stored in the UNIXFILE instance data. The owner and group attributes within the instance are only applied if the Client Automation agent is run as root for only root has the ability to perform changes in owner and group characteristics.
- Client Automation agent capabilities are limited to the permission constraints of the current user ID and group membership for the user ID running the connect.
- Client Automation will be unable to deploy to directories where the directory permissions prohibit the non-root user and or group membership to write.
- Client Automation may be unable to set permissions on resources placed under Client Automation Management that are already on the agent workstation though owned by a different UID and/or GID.
- Client Automation will be unable to launch agent methods requiring root authority.

If the HPCA Application Manager is run as root and:

- If the owner name of the resource, as defined in ZPERUID, and the user ID exist on the agent workstation, the resource will be owned by the user ID specified.
- If the group name of the resource, as defined in ZPERGID, and the group exist on the agent workstation, the resource will be associated with the group specified.



To prevent security breaches please note the following:

If the owner of a resource, as defined in ZPERUID, does not exist on the agent workstation, the owner designation of the managed resource will be set to "nobody" (uid 60001).

If the group of a resource, as defined in ZPERGID, does not exist on the agent workstation, the group designation of the managed resource will be set to "nobody" (gid 60001).

A Modifying Component Instances (Non-Production Only)

At the end of this appendix, you will:

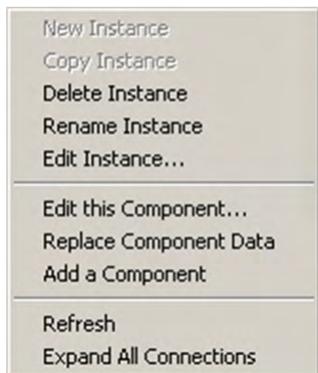
- Understand the restrictions on using the advanced options to Add, Edit, and Replace file-type components.
- Be able to dynamically edit the distributed contents of a file from within the CSDB Editor.
- Be able to replace the entire contents of the data distributed for a file within a published package in the CSDB Editor.
- Be able to add a component instance for a file or behavior to a software package, without having to republish the package.

▶ HP does not support this feature in production environments, and assumes no responsibility if a customer uses this feature where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Client Automation would detect a package change and automatically distribute the change to all authorized agents.

This appendix explains how to dynamically edit the distributed contents of a file from within the CSDB Editor.

Figure 45 below shows the shortcut menu commands for component modifications.

Figure 45 Instance shortcut menu



When to Use these Commands

Examples of when you might want to use this option to modify component instances are listed in the following table.

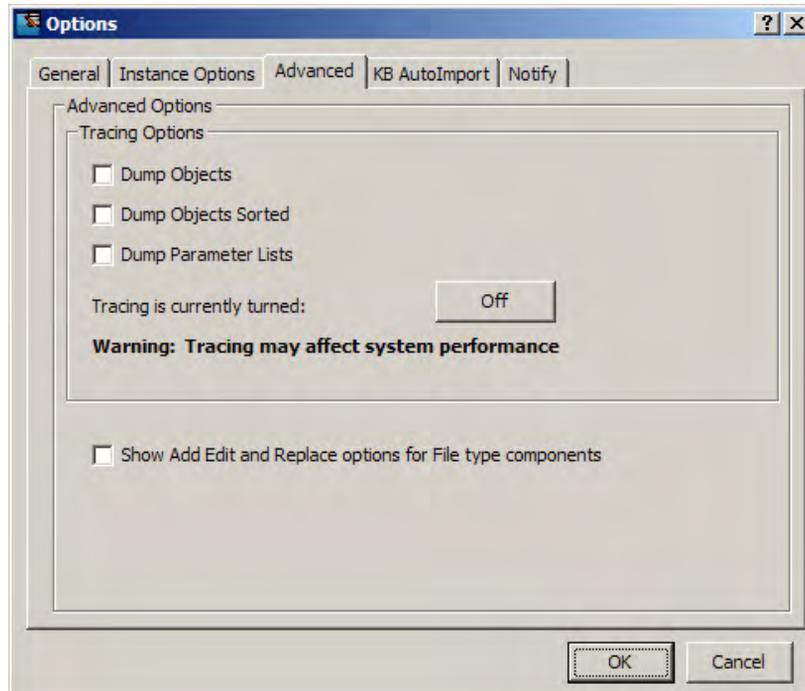
Table 56 When to use the Advanced Edit Component commands

Command	Typical Use
Edit this Component	Change a serial number deployed with a file. Edit the contents of a <code>readme</code> file, document, spread sheet, logo image, or PDF.
Replace Component Data	Replace the entire contents of an existing file in a package with contents of another file (outside of CSDB).
Add a Component	Add a new file instance to a package, using another packaged file as a template.

Accessing the Add, Edit, and Replace Component Commands

The following setting is available on the Advanced tab of the Options dialog box: Show Add Edit and Replace Options for File type components.

Figure 46 Advanced tab of the Options window



For non-production environments only, select **Show Add Edit and Replace options for File type components** to enable these commands on the shortcut menu for applicable instances.

▶ HP does not support this feature in production environments, and assumes no responsibility if a customer uses this feature where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Client Automation would detect a package change and automatically distribute the change to all authorized agents.

When you select this option the Instance shortcut menu opens (see [Figure 45](#) on page 252).

Editing a Component

The **Edit this Component** option permits you to select an editor of your choice with which to dynamically modify the distributed contents of a file instance. After you save the changes to the file, the modified file is sent (re-promoted) to the Configuration Server and the instance in the CSDB is updated.

▶ HP does not support this feature in production environments, and assumes no responsibility if a customer uses this feature where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Client Automation would detect a package change and automatically distribute the change to all authorized agents.

For example, you can edit the distributable contents of any component instance that has been packaged and is represented in the CSDB Editor. You must save the edited file with the same name as the original file. The modified file immediately replaces the existing file in the package. The CSDB Editor takes care of the publishing for you, saving you from having to republish the package.

The original component's contents are not retained in the CSDB Editor.

See [Publisher](#) on page 72 for information on creating and promoting packages to the CSDB.

To edit a component

▶ The component editor does not know a component's target operating system. Before you edit an instance, HP recommends that you note (or export) the values of the following attributes, as well as any other attributes, whose values should not change due to your edits:

ZRSCMMEM (PDS Member Name)

ZRSCCRC (Resource CRC)

ZRSCPADM (PDS AdminID)

ZPUBVER (Packager/Publisher Version)

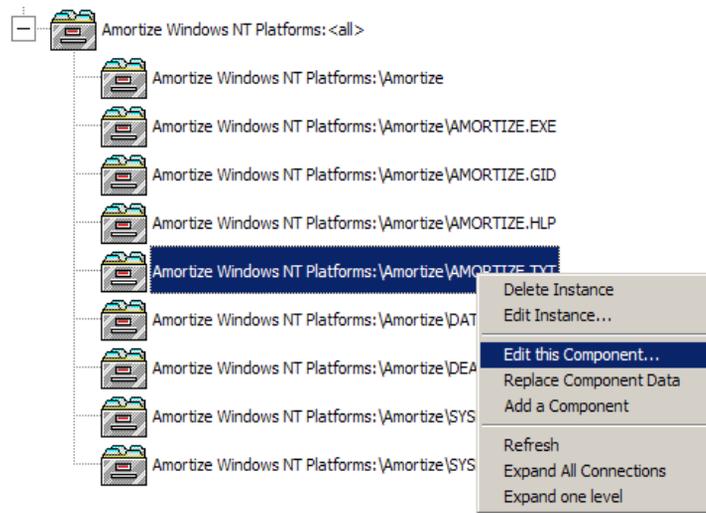
If the values of these attributes change as a result of the editing process, we recommend that you restore the original values.

For example, if a component is targeted for a Windows operating system, the ZRSCMEMM, ZRSCCRC, and ZRSCPADM attributes do not apply and should be null. After editing a component, if these attributes show values other than null, you should clear the inserted values. Likewise, the values for other attributes, such as

ZPUBVER, should remain unchanged due to an edit. If changed, restore ZPUBVER to its original value.

- 1 In the CSDB Editor, go to **Options** → **Advanced** tab, and select **Show Edit Replace and Add Component options for File type components**.
- 2 Locate the file instance that belongs to the software package whose contents you want to change in the CSDB.
- 3 Right-click the instance in the tree view. A shortcut menu opens.

The following figure shows an example of the shortcut menu for the `Amortize.TXT` readme file instance within the Amortize NT Platforms package. It includes the commands: Edit this Component, Replace Component Data, and Add a Component.



- 4 Select **Edit this Component** from the shortcut menu. The Editor Selector for `AMORTIZE.TXT` dialog box opens.

Use this dialog box to select the resident editor you want to use to make the changes to this file. For example, double-click **Notepad** to open the `AMORTIZE.TXT` file with `Notepad.exe`. The Editor Selector dialog box opens.

For details on using this dialog box, see [Using the Editor Selector Dialog Box](#) on page 257.

- 5 Double-click an editor from the Editors list box. The selected editor is launched and opens the previously selected file component instance.

- 6 Make the required changes to the file's contents and save it with the *same name*.
 -  If you save the file with a different name, the Client Automation component editor cannot make the changes to the file you selected from the CSDB Editor.
- 7 Close the editor.
- 8 If necessary, click **OK** to close the Editor Selector dialog box and save the changes to the instance.
- 9 The File Re-Promote dialog box opens, and requests a confirmation that you want to commit the changes to the selected instance in the database.
 - Click **Yes** to save the changes, perform a mini-publish and re-promote the changed file to the database.
 - Click **No** to cancel the edits and the edit session. No changes are made to the database instance.

After you commit the changes, the instance attributes for ZRSCDATE, ZRSCTIME, and ZRSCSIZE reflect the edited file size and date.

File Resources class Amortize Windows NT/2000/XP:\Amortize\AMORTIZE.TXT Instance Attributes:		
Name	Attribute Description	Value
ZRSCNAME	Resource Name	&ZRSCCFIL
ZRSCCFIL	Resource File Name	\Amortize\AMORTIZE.T...
ZRSCMD	Mandatory/Optional on Client [M/O]	0
ZRSCVRFY	Verify Resource File on Connect	Y
ZRSCCONF	Confirm File Download [Y/N]	Y
ZRSCRASH	DOS File Attribute [R/A/S/H]	A
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY
ZRSCMEM	PDS Member Name	21859BAE
ZRSCDATE	Resource Date Stamp - From Promote	20021023
ZRSCTIME	Resource Time Stamp - From Promote	16:27:40
ZRSCSIZE	Resource Size - From Promote	00000001193
ZRSCVERS	Resource Version - From Promote	
ZRSCCSTA	Client File Status	999
ZRSCCKPT	Resource Checkpoint	
ZRSCCRC	Resource CRC	



HP recommends that you compare all attributes' values against their original ones, and restore any values that changed unnecessarily. For example, you might need to clear values from OS-related attributes whose values were originally null. For details, see the cautionary text at the beginning of this procedure on page 254.

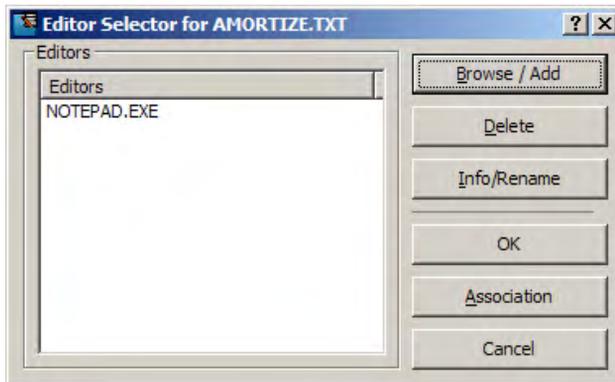
Using the Editor Selector Dialog Box

Use the Editor Selector dialog box to select a resident editor with which to edit the named file.

To select an editor, choose an entry from the Editors list, or click **Association** to use the Windows default editor associated with file's extension.

To add, delete, or rename the editors displayed in the Editors list, use the top three buttons on the right.

Figure 47 Editor Selector dialog box to edit a component



The following table summarizes the Editor Selector dialog box button actions.

Table 57 Editor Selector Dialog Box Buttons

Button	Action
Browse/Add	Opens a File Browse dialog box that you can use to select a program to add to the list of editors. Adds an entry from the Browse dialog box to the Editors list below the highlighted entry.
Delete	Deletes a selected (highlighted) program from the Editors list.

Button	Action
Info/Rename	For the selected editor program, opens a dialog box that displays the editor's program name and path, and allows you to assign a friendly name to display on the Editors list.
OK	Opens the selected file (named in the title bar) with the highlighted editor. Alternatively, you can double-click on the Editors list-box entry.
Association	Opens the selected file (named in the title bar) with the default editor associated with that file type through Windows. If no association is available, you will be returned to the Editor Selector dialog box where you can specify an editor or cancel.
Cancel	Cancel the Edit a Component command without making any changes to the selected component or package.

Replacing Component Data

Use the Replace Component Data command to replace the entire contents of a packaged file with the contents of a file that exists outside the database. For example, if there is a new `Readme.TXT`, and you want to completely replace its contents, prepare a revised `Readme.TXT` file outside of the CSDB Editor. Then, use the Replace Component Data command from within the CSDB Editor to replace the contents of the existing `Readme.TXT` with the new one. When you use the Replace Component Data command, none of the properties of the packaged file changes, just its distributed contents.



HP does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Client Automation would detect a package change and automatically distribute the change to all authorized agents.

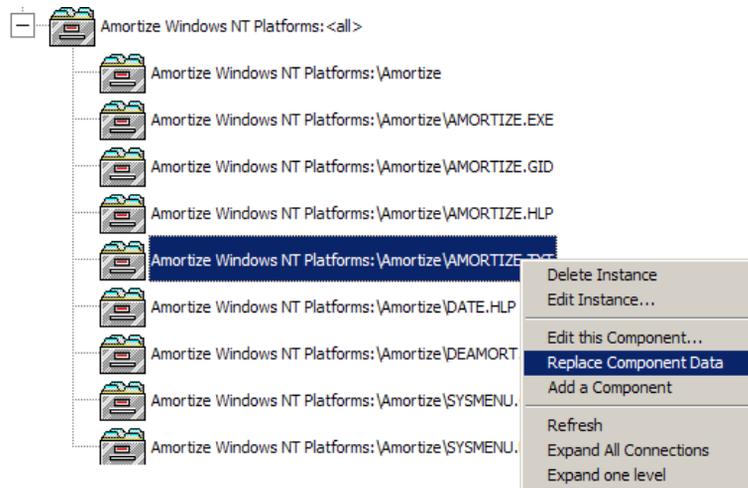
To replace [all] component data

- 1 Prepare a file outside of CSDB Editor whose contents represent the replacement data for a packaged component file in the CSDB. The prepared file can have any name.

► The entire contents of the prepared file will replace the entire data contents of the file selected from the CSDB Editor using the Replace Component Data command. Verify the contents of the files prior to continuing with this procedure.

- 2 In the CSDB Editor, go to **Options** → **Advanced** tab, and select **Show Edit Replace and Add Component options for File type components**.
- 3 Locate the file instance in the software package whose data you want to replace in the CSDB.
- 4 Right-click the instance in the tree view. A shortcut menu opens.

The following figure shows an example of the shortcut menu for the `Amortize.TXT` readme file instance within the Amortize NT Platforms package. It includes the commands: Edit this Component, Replace Component Data, and Add a Component.



- 5 Select **Replace Component Data** from the shortcut menu.
A standard Windows Open dialog box opens.
- 6 Navigate to the prepared file whose contents will replace the data for the file you selected from the CSDB Editor.
- 7 Click **Open** to select the file with the replacement data. A confirmation message opens.
- 8 Click **Yes** to replace the data; click **No** to abort the data replacement.

If you click **Yes**, the CSDB Editor substitutes the entire contents of the data to be distributed with the selected file's data, and republishes the package.

The instance attributes for ZRSCDATE, ZRSCTIME, and ZRSCSIZE reflect the new file size and date, as previously illustrated.

If you click **No**, the replacement command is cancelled.

Adding a Component

The **Add a Component** command is available for file and behavior components only. Files always belong to an owning package, so the most likely place to perform an Add is from the connection under a package. New files can be added by copying the properties of an existing published file; files cannot be created from scratch, that is, from files that have not been promoted from the Packager.

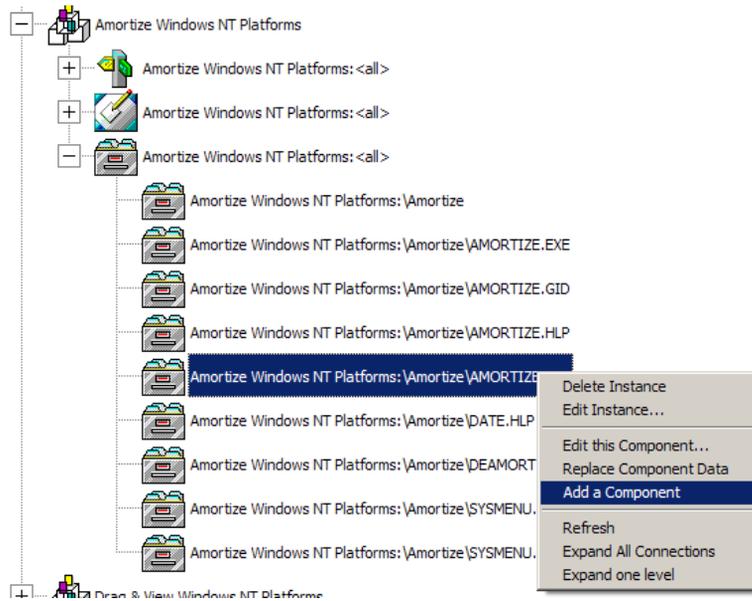
 HP does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are actively being deployed to agents in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Client Automation would detect a package change and automatically distribute the change to all authorized agents.

New files are added by selecting an instance to use as a template. The CSDB Editor will derive the path for the new instance from the template's LOCATION variable. After a template is selected, right-click it to display the shortcut menu and select **Add Component**. An Open File dialog box opens where you can select the file you want to add to your package. A final confirmation box will allow you to stop the process if you need to. Click **Yes** in the box to promote the file that was selected in the Open File dialog box. The package now contains the new file.

To add a component to a package

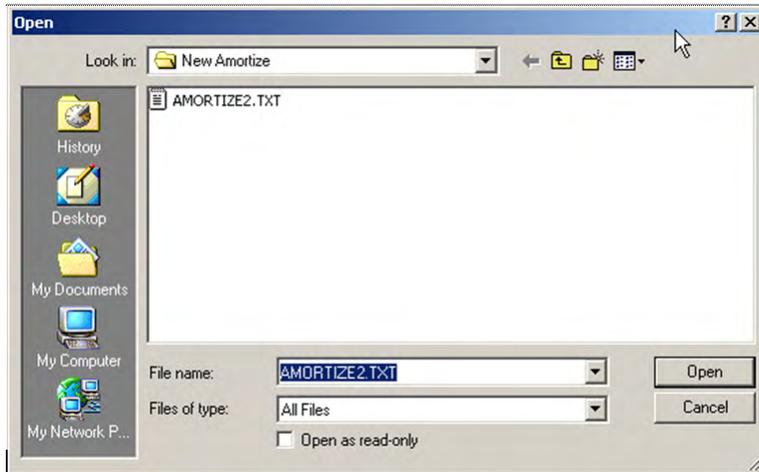
- 1 In the CSDB Editor, go to **Options** → **Advanced** tab, and select **Show Edit Replace and Add Component options for File type components**.
- 2 Locate a file instance in the software package whose LOCATION can serve as a template for the component to be added.
- 3 Right-click the template file's instance in the tree view. A shortcut menu opens.

The following figure shows an example of the shortcut menu for the `Amortize.TXT` readme file instance within the Amortize Windows NT Platforms package. It includes the commands: Edit this Component, Replace Component Data, and Add a Component.



- 4 Select **Add a Component** from the shortcut menu. A standard Windows Open dialog box opens.
- 5 Navigate to the prepared file you want to add to the package.

The following figure shows an example of navigating to a prepared file named `AMORTIZE2.TXT`. In this case, we want to add this component to the package.



- 6 Click **Open** to select the file to be added with the same **LOCATION** as the template file. A confirmation message opens.
- 7 Click **OK** or **Cancel**.

If you click **OK**, the CSDB Editor promotes the added file and republishes the package. The instance attributes for the added file's **LOCATION** reflect the template file's location.

If you click **No**, the Add a Component command is cancelled.

B Naming Conventions

This appendix discusses the use of naming conventions to help you organize the software stored in the Configuration Server Database (CSDB).

When publishing applications, subscribers may have different requirements such as:

- Different operating systems.
- Different amounts of free space on their hard drives.
- Different processors, memory, and so on.
- Different data or applications, depending upon their job function or other factors.

Due to these varying requirements, you might need to create several packages for a single application. To keep your data organized in the CSDB, we recommend that you create a naming convention to be used within your organization.

This section provides some recommendations that you can use as a starting point to create your own standards.

Categorizing Information

In general, consider using unique high-level identifiers with an underscore (_) to categorize information in the CSDB. CSDB Editor groups instances based on the identifier that precedes the underscore.

 If you decide to use a high-level identifier *without* an underscore (_), you can use the CSDB Editor's filtering capabilities to display only the instances with that identifier.

For example, if you had a Windows 2000/XP version and a Windows Server 2003 version of an application to calculate loan amortizations, you might name the packages AMORTIZE_2000/XP and AMORTIZE_2003.

Naming Conventions for the POLICY Domain

We recommend that you use a variation of the following standards. When naming instances in a workgroup, use information that groups your subscribers appropriately. For example, if your company is organized by division and location, you might use conventions such as the following:

Table 58 Naming conventions for the USER Class

Format	Description	Example
USERID	Identifies the subscriber.	SJones
DIV_LOC_DESC	Defines ownership or assignment.	CTS_CLE_EVERYONE
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
DESC	Provides additional description of the group.	EVERYONE (all users)

Naming Conventions for the SOFTWARE Domain

In a company organized by division and location, you might organize your data using the following standards.

Table 59 Naming Conventions for the PACKAGE Class

Format	Description	Example
DIV_LOC_APPNAME_VER_OS	Defines the application.	CTS_CLE_WINZIP_80_XP
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP
VER	Identifies the version of the application.	80

Format	Description	Example
OS	Identifies the operating system that the application runs on.	XP

Table 60 Naming Conventions for Delivery and Auditing Classes*

*All other classes in the SOFTWARE Domain.

Format	Description	Example
REG_DIV_LOC_APPNAME_ VER_OS	Defines the application.	NAM_CTS_CLE_WIN ZIP
REG	Identifies the region.	NAM (North America)
DIV	Identifies the division.	CTS (Corporate Technology Services)
LOC	Identifies the location.	CLE (Cleveland)
APPNAME	Identifies the application.	WINZIP

Determining the conventions that make sense for your organization may take some time. However, creating a convention up front and communicating it to all of your Administrators will keep you organized in the future.

Index

3

32-bit architectures, 23

6

64-bit architectures, 23

A

A New Class is Added check box, 131

Active option

- control files, 66
- files, 65
- folders, 66
- registry keys, 69

Add a Component

- command, 252
- description, 260

Add or Remove Programs application, 219

Add/Remove Programs applet, 220

Additional Files basic publishing option, 99

ADDLOCAL argument, 34

ADDSOURCE argument, 34

addtosvc parameter, 100

Admin CSDB Editor

- control features, 129
- database, changing tree view, 137
- filtering the database, 194
- navigating the tree structure, 137
- opening, 125
- options, 129
 - KB AutoImport tab, 135
 - Notify tab, 136
- shortcut menus, 141

Admin CSDB Editor toolbar buttons, 128

Admin Packager

logging on, 57

Admin Publisher

- logging on, 245
- setting default properties, 47

Administrative Installation Point, 215, 221

Administrator

- installation program, 30
- installing
 - command line, 35
 - Installation Wizard, 32
- removing
 - command line, 37
 - Installation Wizard, 37
- system requirements, 30

Administrator for UNIX

- installing
 - using command line, 236
 - using GUI, 234
- system requirements, 232

Advanced Publishing Mode, 91

- Behaviors, 96
- Features, 94
- Machine/User, 95
- Transforms, 95
- Upgrade Existing, 96

Advanced tab, 134

- options described, 134

ADVERTISE argument, 35

Agent Behaviors

- File Arbitration Method, 55
- File Update/Add Method, 55
- Instance Update Method, 55
- Method to De-install Resource, 55

Agent Behaviors tab in Installation Monitor Mode, 53

Agent Management tab

- delivery options, 50

- Installation Monitor Mode, 48
 - verification options, 49
- Agent properties
 - execute, 149
 - protect, 149
- AIP, 30
- aiparguments setting, 101
- aiplocation setting, 101
- alias name, 152
- always connection, 147
- Application Packages instance, description, 21
- AppSync, 88
- ARBITRAT attribute, 55, 80
- architectures, 23
- Assignment type, 86
- Association button, 258
- ASYCFILT.DLL, 57
- attr CLASS {body} parameter, 101
- Attribute Contains the Instance Alias Name, 152
- attribute type
 - connection, 147
 - expression, 147
 - method, 148
 - variable, 147
- attribute, alias, 152
- Automatic management type, 86
- Availability window in Installation Monitor Mode
 - After check box, 60
 - Before check box, 60
- Available resources per user, 95

B

- base instance, 198
- Basic Publishing Mode, 98
 - Additional Files, 99
 - Management Options, 98
 - Transforms, 99
 - Upgrade, 99

- Batch Mode, 103
- Batch Publisher, 26
 - batchmsi parameter, 103
- Behaviors advanced publishing option, 96
- Browse/Add button, 257
- byte level differencing patching, 178

C

- Cache location, 97
- Cancel button, 258
 - cfg parameter, 100, 103
- Check for existence only
 - Component Select mode, 78
 - Installation Monitor Mode, 49
- Check internal version, Installation Monitor Mode, 49
- class
 - adding
 - attribute, 151
 - components, 260
 - instance, 156
 - attribute types
 - connection, 147
 - expression, 147
 - method, 148
 - variable, 147
 - connection types, 147
 - copying, 142
 - creating new, 144
 - deleting, 155
 - editing, 145
- class connection, 147
- Class Editor, 142
- class information, properties, 145
- class type
 - component, 147
 - configuration, 147
 - none, 147
 - policy, 147
 - values, 147
- clean computer, definition, 30

- client method command lines, 54
- COMCAT.DLL, 57
- COMCTL32.OCX, 57
- COMDLG32.OCX, 57
- command line
 - defining file to publish in Windows, 103
 - installing
 - Administrator for UNIX, 236
 - Administrator for Windows, 35
 - publishing in Windows, 100
 - removing Administrator for Windows, 37
- component
 - adding to a package, 260
 - editing, 254
- component class instances, 198
- component classes, 198
 - filtering, 198
- component data, replacing, 258
- component editor, 254
- Component Select
 - description, 21
- Component Select mode
 - delivery options, 78
- Component Select Mode
 - overview, 244
 - verification options, 77
- components, adding, 260
- compression settings
 - Deflate, 51
 - None, 52
- compression settings in Installation Monitor Mode, 51
- configuration file
 - multiple, 104
 - using, 100
- Configuration Server Database
 - adding
 - class, 142
 - copying, 142
 - class attribute, 151
 - components, 260
 - domain, 142
 - instance, 156
 - copying, 156
 - from scratch, 156
 - changing tree view, 137
 - Configuration Server Properties table, 149
 - deleting a class, 155
 - deleting an instance, 161
 - drop down lists, 153
 - editing, 141
 - class, 145
 - instance, 157
 - filtering, 194
 - flag attribute, 152
 - flag set attribute, 153
 - navigating the tree structure, 137
 - navigation history
 - discarding, 138
 - Prune Below, 138
 - new class, creating, 144
 - optional attribute editing, 152
 - renaming an instance, 161
- Configuration Server properties, table, 149
- connection attribute type, 147
- connection rule variable, 208
- connection rules, 202
- connections, enabling, 206
- Content (CRC check)
 - Component Select Mode, 78
 - Installation Monitor Mode, 49
- control files, editing, 66
- copying package, 222

D

- data options
 - Compression settings, 51
 - Promote instances without data, 52
 - Promote Resource As, 52
 - Enforce System File Protection, 52
 - Force lock method, 52

- Maintenance, 52
 - Normal, 52
- data options tab in Installation Monitor Mode, 51
- DATA1.MSI, 219
- date and time constraints, 60
- default properties, setting, 47
- Deflate compression setting, 51
- Delete (Default), 227
- Delete Always, 226
- Delete button, 257
- Delete Never, 226
- deleteaip setting, 101
- delivery options
 - Component Select Mode, 78
 - Machine, 50, 68
 - Mandatory, 50, 78
 - Optional, 50, 79
 - Override Priority, 50, 79
 - Use default priority, 50, 79
 - User, 50, 68
 - User Specified, 51
- Desktop instance, description, 22
- Desktop tab in Installation Monitor Mode, 67
- DHCP. *See* Dynamic Host Configuration Protocol
- Directories to be Scanned list, 62
- DIRPERMS, 248
- Disable feature, 217
- Disk Cost, 34
- DISPLAY attribute, 232
 - setting, 233
- document changes, 4
- domain, adding to a file, 142
- Dynamic Host Configuration Protocol, 56

E

- Edit Instance dialog box, 133
- Edit this Component

- command, 252
- description, 254
- editing a component, 254
- Editing Class dialog box, 131, 145
- Editing Instance dialog box, 157
- Editor Selector dialog box, 255
 - buttons, 257
 - using, 257
- Editors list box, 255
- Email notification, 193
- Enable per user interface, 95
- enableperuser setting, 101
- enforce system file protection
 - in Installation Monitor Mode, 52
 - when publishing files, 52
- execui setting, 101
- Exist/Exist option, 69, 82, 225
- Exists verify option for desktop items in Installation Monitor Mode, 68
- express filters, 199
 - custom, 202
 - customizing, 200
 - description, 200
- expression attribute type, 147

F

- Feature Set Editor
 - Add or Remove Programs, 219
 - Add/Remove Programs, 220
 - buttons, 218
 - Disable feature, 217
 - Hide feature, 217
 - navigating the tree, 213
 - shortcut menu, 212
- Features advanced publishing option, 93
- file
 - adding a domain to, 142
 - editing properties, 65
 - setting default properties, 47

- File Arbitration Method
 - Component Select, 80
 - Installation Monitor Mode, 55
- File Content Comparison check box, 61
- File Contents to be Scanned list, 61
- FILE instance, description, 21
- file parameter, 103
- File Resources instance, description, 21
- File Update/Add Method
 - Component Select, 80
 - Installation Monitor Mode, 55
- Files and Progress information, 183
- Files tab
 - Installation Monitor Mode, 65
- Files tab in Installation Monitor Mode, 64
- Filter Instances dialog box, 195, 198
- filter results, appending, 198
- filter specifications
 - description, 194
 - examples, 194
 - valid expressions, 194
- filter, removing, 65
- filtering, 194
- filtering files, 64
- filters
 - express, 199
 - permanent
 - deleting, 197
 - description, 195
 - setting, 195
 - temporary
 - deleting, 198
 - description, 197
 - setting, 197
- Force lock method in Installation Monitor Mode, 52
- Full Verify option for desktop items
 - Installation Monitor Mode, 68

G

- General tab, 130
- Global Default Properties dialog box, 48
- global defaults, changing, 48

H

- hardware configuration elements, 108
 - publishing, 108
- Hide and Disable shortcut menu, 217
- Hide feature, 217
- HPCA Core, 15
- HPCA Satellite, 15
- HPCAIE-ADMIN72.MSI, 31, 36

I

- IDMLIB directory, 236
- IDMLOG, 236
- Ignore feature, 217
- Inactive option
 - control files, 66
 - files, 65
 - folders, 65
 - registry keys, 69
- includes connection, 147
- Info/Rename button, 258
- installation methods for Administrator
 - UNIX, 233
 - Windows, 30, 36
- Installation Monitor Mode
 - description, 21
 - installing the software, 62
 - Instance Properties dialog box
 - Agent Behaviors tab, 53
 - Agent Management tab, 48
 - Data Options tab, 51
 - overview, 47
 - prerequisites, 56
 - Set Drives to Scan window, 62
 - Set Properties and Locations window, 65

- Desktop tab, 67
- Registry tab, 68
- System Configuration window, 59
- updated files, 66
- installation wizard
 - installing Administrator for Windows, 32
 - removing Administrator for Windows, 37
- Installed on First Use feature, 216
- installing
 - Administrator
 - command line, 35
 - Installation Wizard, 32
 - Administrator for UNIX
 - using command line, 236
 - using GUI, 234
 - software in Installation Monitor Mode, 62
- instance
 - adding, 156
 - adding from scratch, 156
 - copying, 156
 - deleting, 161
 - editing, 157
 - renaming, 161
- instance names, prefixes, 139
- Instance Options tab, 132
 - options described, 132
- instance properties
 - agent behaviors
 - File Arbitration Method, 55
 - File Update/Add Method, 55
 - Instance Update Method, 55
 - Method to De-install Resource, 55
 - data options
 - Compression setting
 - Deflate, 51
 - None, 52
 - Promote instances without data, 52
 - Promote Resource As, 52
 - Enforce System File Protection, 52
 - Force lock method, 52
 - Maintenance, 52
 - Normal, 52

- delivery options
 - Machine, 50
 - Mandatory, 50, 78
 - Optional, 50, 79
 - Override Priority, 50, 79
 - Use default priority, 50, 79
 - User, 50
 - User Specified, 51
- verification options
 - Check for existence only, 49, 78
 - Check internal version, 49
 - Content (CRC check), 49, 78
 - No verification, 50, 78
 - Update only if newer, 49, 78
 - Use the default specified on the Manager, 49, 77
 - Verify statistics equal to, 49, 77
- Instance Properties dialog box, 66
 - Installation Monitor Mode
 - Agent Behaviors tab, 53
 - Agent Management tab, 48
 - Data Options tab, 51
- Instance Update Method
 - Component Select, 80
 - Installation Monitor Mode, 55
- instances
 - prefixed names, 139

J

- Just In Time, 67

K

- KB AutoImport tab, 135

L

- LOCATION variable, 260

- Lock the splitter bar distance to the right side of the screen, 132

- locked file method, 52

- log directory, 104

- loglevel 9, 233

- logon dialog box, 125

M

- Machine / User publishing option, 95
- machine delivery option
 - Installation Monitor Mode
 - desktop items
 - files, 50
- Maintenance data option in Installation Monitor Mode, 52
- maintenance files, promoting, 52
- Managed applications, 97
- Management Options basic publishing option, 98
- Management type, 86
- Mandatory assignment type, 86
- Mandatory delivery of files
 - Component Select mode, 78
 - Installation Monitor Mode, 50
- Manual management type, 86
- master service, 188
 - creating, 189
- MD5 signature, 178
- memory required
 - Component Select, 85
 - Installation Monitor Mode, 60
- message-digest algorithm, 177
- method attribute type, 148
- Method to De-install Resource
 - Component Select, 80
 - Installation Monitor Mode, 55
- Method to Install Resource
 - Component Select, 79
- msi {body} parameter, 101
- MSI substitution value sources, 107
- MSIEXEC, 223
- MSIFEATS, 212, 222, 223
- mst files, 103, 222
- MSVBVM60.DLL, 57
- multiple instances

Index

- editing, 158
- Editing Instance dialog box, 159
- selecting, 158

N

- Name Lists Class, 204
- naming conventions, 17, 263
 - POLICY Domain, 264
 - SOFTWARE Domain, 264
- Native Mode, 222
- Native Packaging, 27
- New Key, 226
- New Value, 226
- No verification of files
 - Component Select mode, 78
 - Installation Monitor Mode, 50
- No Verify option for desktop items
 - Installation Monitor Mode, 68
- None compression setting, 52
- Normal data option in Installation Monitor Mode, 52
- Not Available feature, 216
- notify
 - by machine name, 136
 - description, 192
 - initiating, 194
 - specify STARTDIR, 136
 - specify UID, 136
- Notify tab, 136
- null instance, 198

O

- OK button, 258
- OLEAUT32.DLL, 57
- OLEPRO32.DLL, 57
- Open Packaging Session window, 58
- Operating System list box, 59
- operating system required
 - Component Select, 85

- Installation Monitor Mode, 60
- Optimize Service, 178
 - configuring, 179
 - using, 178
- Optional assignment type, 86
- Optional delivery of files
 - Component Select Mode, 79
 - Installation Monitor Mode, 50
- Override Priority delivery of files
 - Component Select mode, 79
 - Installation Monitor Mode, 50
- Override Value, control files, 66
- owning package, 199

P

- package
 - definition, 20
 - duplicating, 222
 - promoting, 70, 87
- PACKAGE Class instance, 59
- package information
 - domain
 - Component Select, 84
 - Installation Monitor Mode, 59
 - package description
 - Component Select, 85
 - Installation Monitor Mode, 59
 - package name
 - Component Select, 84
 - Installation Monitor Mode, 59
 - release number
 - Component Select, 85
 - Installation Monitor Mode, 59
- PACKAGE Instance, description, 21
- package names, 6, 103, 104, 106
- package parameter, 100
- packaging, definition, 20
- passport registration, 8
- patch building, 184
- PATCH Class, 178, 186
 - Patch Criteria
 - button functions, 185
 - information, 184
 - PATCH Domain, 178, 186
 - patch-building restrictions, 184
 - Path instance, description, 22
 - path parameter, 101
 - permanent filters, 195
 - deleting, 197
 - description, 195
 - setting, 195
 - pkgdesc parameter, 100
 - pkgname parameter, 100, 106
 - plain mode for installing Administrator for UNIX, 234
 - Populate Windows Installer Methods, 172
 - prefixes, using in compound names, 139
 - prerequisites, 16
 - Preserve internet cache, 97
 - progress bar, displaying, 36
 - promote
 - a package, 70, 87
 - description, 21
 - instances created during, 21
 - promote instances without data in Installation Monitor Mode, 52
 - Promote Resource As
 - Enforce System File Protection, 52
 - Force lock method, 52
 - Installation Monitor Mode, 52
 - Maintenance, 52
 - Normal, 52
 - Properties advanced publishing option, 94
 - Properties option, registry keys, 69
 - Prune Below, 138
 - Publish with AIP, 93
 - publishing
 - Component Select

- description, 21
- overview, 73
- Installation Monitor Mode
 - description, 21
 - overview, 47
- multiple .msi files, 103
- overview for UNIX, 244
- with command line in Windows, 100

publishmode setting, 101

publishr executable, 245

pubport.log file, 104

R

Radreg32 Delete operation, 226

radskman, 188

REG_DWORD, 226

REG_EXPAND_SZ, 226

REG_MULTI_SZ, 226

REG_SZ, 226

registry

- editing values, 69
- restoring to original value, 69
- setting properties, 69
- viewing, 68

Registry check box, 61

Registry File Editor, 223

- accessing, 223
- closing, 229
- setting properties, 225
- shortcut menu options, 225

REGISTRY instance, description, 22

registry key, editing, 227

Registry Resources instance, 223

- description, 22

Registry tab in Installation Monitor Mode, 68

related documents, 17

REMOVE argument, 35

Remove option, 37

removing Administrator for Windows

command line, 37

Installation Wizard, 37

Rename Instance dialog box, 161

Replace Component Data

- command, 252
- description, 258

requires connection, 147

Resource Initialization Method

- Component Select, 79

Run all from My Computer feature, 215

Run all from Source feature, 215

Run from My Computer feature, 215

Run from Source feature, 215

S

Salvage, 97

scanning

- identifying what to scan, 61
- the file system, 62

security dialog box, 125

Selected Files, viewing, 64

Selected Links, 67

Selected Registry Keys, 68

Service Groups, 187

- master service, 188
- prerequisites, 188

Service Optimization, 177

service parameter, 100

service, creating, 170

Session Type in Installation Monitor Mode, 58

Set Drives to Scan window, 62

Set Properties option

- files, 65
- folders, 66

setting file properties

- Component Select, 76
- Installation Monitor Mode
 - Desktop tab, 67

- Files tab, 65
- Registry tab, 68
- setup.exe, 30
- SFP. *See* System File Protection
- shortcut menus, 141
- SHOWINCP variable, 219, 220
- silent installation, 36
- STDOLE2.TLB, 57
- svcdesc parameter, 100
- svcname parameter, 100
- symbolic substitution, 106
- System Configuration window
 - Component Select, 85
 - Installation Monitor Mode, 59
 - Minimum Memory Required, 60
 - Minimum Processor Required, 60
- System File Protection
 - enforcing during promote, 52
 - Installation Monitor Mode, 52
- system requirements for Administrator
 - UNIX, 232
 - Windows, 30

T

- target directory, 31, 36
- Target Operating System
 - Component Select, 85
 - New Application Wizard, 172
- temporary filters
 - deleting, 198
 - description, 197
 - setting, 197
- text mode for installing Administrator for UNIX, 234
- ThinApp, 88
- transform file, 222
- transform files, 103
- Transforms advanced publishing option, 94
- Transforms basic publishing option, 99

- transient classes, 146
- tree view, manipulating, 137
- Type menu commands for Installation Monitor Mode, 68

U

- Unconditional/Enforced option, 69, 81, 225
- Unconditional/Exist option, 69, 81, 225
- UNIX File Resources, 247
- UNIXFILE attributes, 247
- UNIXFILE Class, 248
- UNIXFILE instance, 247
- UNIXFILE.ZRSCRASH attribute, 248
- Unmanaged applications, 97
- Update only if newer
 - Component Select mode, 78
 - Installation Monitor Mode, 49
- Updated files, Installation Monitor Mode, 66
- Upgrade basic publishing option, 99
- Upgrade publishing option, 95
- Use Base, 87
- Use default priority delivery of files
 - Component Select Mode, 79
 - Installation Monitor Mode, 50
- Use local cache for MSI files, 97
- Use Local Repair, 97
- Use long filenames check box, 93
- Use the default verification specified on the Manager
 - Component Select Mode, 77
 - Installation Monitor Mode, 49
- User controls MSI features, 98
- User delivery option
 - Installation Monitor Mode
 - desktop items, 68
 - files, 50
- user specified delivery option in Installation Monitor Mode, 51

V

- variable attribute type, 147
- verification options
 - Check for existence only, 49, 78
 - Check internal version, 49
 - Component Select Mode, 77
 - Content (CRC check), 49, 78
 - Exists, 68
 - Full Verify, 68
 - No verification, 50, 78
 - No Verify, 68
 - Update only if newer, 49, 78
 - Use the default specified on the Manager, 49, 77
 - Verify statistics equal to, 49, 77
- Verify statistics equal to
 - Component Select Mode, 77
 - Installation Monitor Mode, 49
- Version Group Editor, description, 177
- VMware, 88

W

- Windows Installer AIP, creating, 36
- Windows Installer files, 88
- Windows Installer log, 36

Z

- ZCONTEXT attribute, 50
- ZCREATE attribute, 79
- ZDELETE attribute, 55, 80
- ZERROR instance, 53
- ZFILEUPD attribute, 55, 80
- ZINIT attribute, 79
- ZLIST Class, 203
- ZOBJPTCH variable, 178
- ZPERGID attribute, 248, 249
- ZPERUID attribute, 248, 249
- ZPUBVER, 254
- ZRSCCRC, 254
- ZRSCMMEM, 254
- ZRSCMO attribute, 50
- ZRSCPADM, 254
- ZRSCPRI attribute, 50
- ZRSCRASH attribute, 248
- ZRSCVRFY attribute, 49, 52, 53, 78
- ZSVCGRP variable, 189
- ZSVCPRI, 192
- ZUPDATE attribute, 55, 80

