

HP OpenView Administrator Workstation Using Radia

Radia System Explorer Guide

Version: 4.0i

for the Windows® operating system



Manufacturing Part Number: T3420-90015

July 2005

© Copyright 2004 Hewlett-Packard Development Company, L.P.

Legal Notices

Warranty

Hewlett-Packard makes no warranty of any kind with regard to this document, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

A copy of the specific warranty terms applicable to your Hewlett-Packard product can be obtained from your local Sales and Service Office.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
United States of America

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19(c)(1,2).

Copyright Notices

© Copyright 1998-2004 Hewlett-Packard Development Company, L.P.

No part of this document may be copied, reproduced, or translated into another language without the prior written consent of Hewlett-Packard Company. The information contained in this material is subject to change without notice.

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

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

Technical Support

Please select Support & Services from the following web site:

[<http://www.hp.com/managementsoftware/services>](http://www.hp.com/managementsoftware/services)

There you will find contact information and details about the products, services, and support that HP OpenView offers.

The support site includes:

- Downloadable documentation
- Troubleshooting information
- Patches and updates
- Problem reporting
- Training information
- Support program information

Preface

About this Guide

Who this Guide is for

This guide should be used by systems administrators who want to create and maintain a distribution model for their enterprises.

What this Guide is about

This book introduces you to the Radia System Explorer—the Radia tool you use to configure and administer your enterprise desktop environment. It describes the concepts of maintaining a distribution model in the Radia Database, and the actions you take when administering the Radia environment.

Summary of Changes

This printing of the *Radia System Explorer Guide for Windows* for use with Radia version **4.0** contains the following changes to information and procedures for the following chapters.

4.0 Note

Items with **4.0** bullet are specific to version **4.0**.

Chapter 2: Getting Started

- 4.0** Page 32, *Setting Preferences: The Options Dialog Box*. Added Notify to the list of tabs on the View Options dialog box.
- 4.0** Page 34, *Table 2.7 ~ Settings on the General Tab and their Functions*, added a row for new window resizing option, Lock the splitter bar distance to the right side of the screen.
- 4.0** Page 40, Notify Tab: new section. The **Notify** tab on the **Options** dialog box enables customization of the UID and STARTDIR values in the command line generated for the Notify Subscribers. Select the **Custom** option to notify clients by machine name, or specify a starting directory.

Chapter 5: Additional Features of Radia System Explorer

- 4.0** Page 180, *Set Notification Details*. This dialog box now displays the values for UID and STARTDIR that will be used with the Notify command. These values can be changed on the **Notify** tab of the **Options** dialog box.

Editorial Improvements

In addition to the changes listed above, this version contains various editorial and style updates to each chapter and section and the index.

Conventions

You should be aware of the following conventions used in this book.

Table P.1 ~ Styles

Element	Style	Example
References	<i>Italic</i>	See the <i>Publishing Applications and Content</i> chapter in this book.
Dialog boxes and windows	Bold	The Radia System Explorer Security Information dialog box opens.
Code	Andale Mono	radia_am.exe
Selections	Bold	Open the \Admin directory on the installation CD-ROM.

Table P.2 ~ Usage

Element	Style	Example
Drives (system, mapped, CD)	Italicized placeholder	<i>SystemDrive</i> : \Program Files\HP might refer to C:\Program Files\HP on your computer. <i>CDDrive</i> : \client\radia_am.exe might refer to D:\client\radia_am.exe on your computer.
Files (in the Radia Database)	All uppercase	PRIMARY
Domains (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE May also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE May also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

The table below describes terms that may be used interchangeably throughout this book.

Table P.3 ~ Terminology*

* Depends on the context. May not always be able to substitute.

Term	May also be called
Application	software, service
Client	Radia Application Manager and/or Radia Software Manager
Computer	workstation, server
NOVADIGM domain	PRDMAINT domain Note: As of the 4.0 release of the database, the NOVADIGM domain is being renamed the PRDMAINT domain. Therefore, if you are using an earlier version, you will see the NOVADIGM domain in the database.
Radia Configuration Server	Manager, Active Component Server
Radia Database	Radia Configuration Server Database

Contents

Preface	5
About this Guide	5
Who this Guide is for	5
What this Guide is about	5
Summary of Changes	6
Conventions	7
 Introduction	 15
About the Radia System Explorer	16
The Distribution Model	17
Radia Terminology	17
Handling Your Unique Requirements	18
Summary	20
 Getting Started.....	 21
Using a Mouse to Manipulate Elements in the Window.....	22
Opening the Radia System Explorer.....	22
The Radia System Explorer Window	25
Basic Windows Operations	25
Customizing the Radia System Explorer Window.....	25
Radia System Explorer Window Menus	26
The Radia System Explorer Toolbar.....	31
Setting Preferences: The Options Dialog Box.....	32

General Tab.....	34
Instance Options Tab	35
Advanced Tab.....	37
KB AutoImport Tab	39
Notify Tab	40
Summary	42

Radia Database Overview 43

The Radia Database.....	44
Navigating through the Radia Database	46
Navigation History.....	48
Compound Names.....	50
Inheritance in the Radia Database	52
Radia System Explorer Shortcut Menus	53
Working with Domains	54
Adding a Domain	54
Working with Classes	55
Adding a Class	55
Copying a Class.....	55
Create a New Class from Scratch.....	57
Editing a Class	61
Client Properties	64
Manager Properties	65
Adding an Attribute to a Class	66
Optional Attribute Editing Controls.....	67
Deleting a Class	72
Working with Instances.....	73
Adding an Instance	73
Copying an Instance.....	73
Adding an Instance from Scratch.....	74
Editing an Instance	75
Editing Multiple Instances.....	77
Renaming an Instance.....	82
Deleting an Instance	82
Summary	84

Building a Distribution Model for Your Radia Database 85

Building a Distribution Model.....	86
The Policy Domain	86
Software Domain	91
Inter-Package Relationships.....	94
Determining Policies	97
Connection Types	100
Manual Connections	100
Drag-and-Drop Connections.....	100
Show Connections.....	102
Drag-and-Drop Component Class Instances.....	110
Connecting Services to Policies.....	112
A Word about the Radia Policy Server	116

Additional Features of the Radia System Explorer 119

Running the New Application Wizard	120
Defining MSI Basic Behaviors on the New Application Wizard	132
The Version Group Editor.....	134
Creating a Version Group Instance.....	134
Version Group Assignments	141
Service Optimization.....	145
Using the Optimize Service Option	147
Radia Service Groups.....	165
Component Enhancements.....	165
Prerequisites	165
Creating a Master Group	167
Radia Notify	174
Notify Dialog Boxes.....	176
Send an Email Using Notify.....	179
Update an Application Using Notify	179
Remove an Application Using Notify	179
Set Notification Details	180
Notify File Structure	186
Retrying a Notify Operation	186
Drag-and-Drop Notify for Wake-On-LAN Clients.....	187
Filtering—Viewing Portions of the Radia Database.....	187
Permanent Filters	188
Temporary Filters	192

Viewing the Base or Null Instance	192
Appending Filter Results	193
Filtering Component Class Instances	193
Express Filters	194
Customizing Express Filters	195
Summary	198
Understanding the Resolution Process	199
The Resolution Process	200
Heaps and Objects	200
Resolution Process – Where Does It Begin?	203
Resolution Example	205
Resolution Process and Symbolic Substitution	206
Summary	210
Maintaining Connection Rules in the Radia Database.....	211
Introduction	212
Connection Rules Overview.....	212
Enabling Connections in Newly Added Classes	216
Summary	222
Feature Set Editor	223
Feature Set Editor Overview	224
Accessing the Feature Set Editor.....	225
About the Radia MSI Feature Set Editor	226
Tree View	227
Features Menu	229
Current Feature Settings.....	233
Hide and Disable Features	235
Feature Set Editor Buttons.....	237
Package Information	238
Maximum Size.....	238
Total Size.....	238
Selected Size.....	238
Windows Add/Remove Programs Applet.....	239
Copying a Windows Installer Enabled Package	241

Managing Feature Sets in Native Mode	245
Summary	246
Registry File Editor	249
The Registry File Editor	250
Accessing the Registry File Editor	250
Expanding and Collapsing the Tree	253
Set Properties Context Menu	254
Editing a Registry Key	256
Closing the Registry File Editor	260
Summary	261
Support for the Radia Inventory Manager	263
About the Radia Inventory Manager	264
Overview	264
Radia and WBEM	265
The Radia Integration Server	265
Summary	266
Radia System Explorer Shortcut Menus	267
Radia System Explorer Shortcut Menus	267
Modifying Component Instances (Non-Production Only)	283
When to Use these Commands	284
Accessing the Add, Edit, and Replace Component Commands	285
Editing a Component	286
Using the Editor Selector Dialog Box	291
Replacing Component Data	292
Adding a Component	296
Summary	300
Lists	301
Figures	301

Contents

Tables.....	310
Procedures.....	312
Index.....	315

Introduction

At the end of this chapter, you will:

- Be familiar with the Radia System Explorer and its uses.
- Understand the Radia distribution model.
- Be familiar with Radia terminology.

About the Radia System Explorer

The Radia System Explorer is an interactive graphical tool for manipulating and inspecting the content of the Radia Configuration Server Database.

By working in the Radia System Explorer, you can:

- Define users (i.e., identify desktop computers to be managed by Radia) in the Radia Database.
- Assign applications to individual users and workgroups.
- Define applications and their files.
- Define special file properties.
- Group applications based on your organization's policies.
- Grant users access to applications, or to groups of applications.
- Manage the configuration of users' desktops.

Radia enables you to configure applications, connect users to applications, and define hardware and software auditing requirements for desktop computers throughout your enterprise. This will be based on the policies that your organization sets for its desired configurations.

The Distribution Model

Your *distribution model* records the identities and intended configurations of the desktop computers whose relationships are managed by Radia. It can be as sophisticated, or as simple, as you want. At a minimum, a Radia distribution model includes the following five elements:

- **Users**
The identity of the desktops being managed.
(Example: a user's user ID)
- **Applications**
The software that is being managed.
(Example: Microsoft Office)
- **Application Files**
The components that make up the application.
(Example: Microsoft Office has approximately 800 files, DLLs, EXEs, HLPs, ICOs, as well as hundreds of registry updates)
- **Deployment Source**
The location where the application components are stored so they can be deployed to the users.
(Example: the Radia Database, or a local Radia Staging Server)
- **Deployment Destinations**
The location to which the application and its files will be deployed on the desktop or LAN (local area network).
(Example: C:\MSOFFICE\, C:\WINDOWS\)

The remaining chapters of this book explain how to use the Radia System Explorer to configure and maintain distribution models for your applications.

As you gain experience with Radia and become a more advanced user, you will probably want to include other elements in your distribution models. A few of these capabilities include distribution scheduling, error handling, security, and collecting audit information from desktops. Refer to the Radia library for descriptions of these concepts and capabilities.

Radia Terminology

The following terms will be explained in detail later, but you should become familiar with these key concepts now.

The Radia Database records your distribution model and information regarding its deployment. The Radia Database is hierarchically structured, and its components consist of **Files**, **Domains**, **Classes**, **Instances**, and **Attributes**. See Table 1.1 on page 18 for additional information.

Table 1.1 ~ Key Radia Terms

Term	Short Description
File	Highest level in the Radia Database hierarchy. Groups similar domains together.
Example: PRIMARY file	The PRIMARY file is used to define and maintain the distribution model. This is a pre-configured files distributed with Radia and is installed when you first install Radia.
Domain	Logically partitions a Radia file. Groups similar classes together.
Example: SYSTEM domain	The SYSTEM domain contains the classes needed to deploy applications.
Class	A category of the distribution model. The class is a template for the attributes (also called <i>properties</i> or <i>fields</i>) needed to create an instance of the class. It is conceptually similar to a schema in a relational database structure, or a file layout in a traditional flat file. Each of the five required elements of a distribution model (users, applications, application files, deployment sources, and deployment destinations) is defined in the Radia Database by its class. Refer to the <i>Radia Database Reference Manual</i> for information on the structure and usage of Radia classes.
Example: USER class	The USER class defines users of Radia-managed applications. The USER class defines all the attributes necessary to identify a desktop (client) computer.
Class Instance or Instance	An object containing a specific occurrence of a class. This is analogous to a row in a relational data table, or a record in a traditional flat file. The attributes of a class instance object contain data describing one specific entity of that class.
Example: USER class instance defining John Doe's computer	This USER instance is an object created from the USER class, containing the information needed to identify John Doe's Radia-managed computer.
Attribute	An attribute is a data element (also called a <i>field</i> or <i>property</i>) of a class. The class contains the definition (e.g., the name, data type, description, and length) for each attribute comprising the class. Each class instance created from the class contains a value for each of the attributes defined in the class.
Attribute Value	
Example: USER class instance defining John Doe's computer, NAME and USERID attributes	The NAME attribute of the USER class contains the name of the user, and the USERID attribute contains the user ID, as specified by the Radia administrator. In this example, the NAME attribute contains the value "John Doe", and the USERID attribute contains the value "JDOE".

Handling Your Unique Requirements

Radia ships with the Radia Database pre-configured with a set of components that can be used for managing desktop computer configurations on an enterprise-wide scale. Using these components, you can build and maintain very sophisticated and complex distribution models.

However, it is important to note that Radia provides a highly customizable framework that you can modify and extend to meet your organization's individual requirements. For example, it is easy to extend the basic functionality of Radia by adding your own components to the Radia Database, extending Radia-supplied components, and integrating your own in-house developed or third-party-supplied executables.

In particular, this document describes the Radia System Explorer using the Radia Database as it is configured when shipped from HP. If your organization customizes Radia, the Radia System Explorer can be used as described here, but the sample screens shown in this document may differ from what you see in your live environment.

For purposes of maintainability, customization of Radia should be thoroughly documented and diligently kept up to date in a central location. You might want to create a project folder that is accessible to all of your Radia administrators.

Summary

- The Radia System Explorer is an interactive graphical tool for manipulating and inspecting the content of the Radia Database.
- The Radia Database records your distribution model and information regarding its deployment.
- Your distribution model records the identities and intended configurations of the desktop computers whose relationships are managed by Radia.
- Although Radia ships with a pre-configured database, the framework is highly customizable.

Getting Started

At the end of this chapter, you will:

- Know how to use a mouse to manipulate Radia System Explorer elements.
- Know how to open the Radia System Explorer.
- Manipulate the appearance of an open Radia System Explorer window.
- Know the Radia System Explorer menus.
- Know how to set your personal preferences.
- Know how to set the defaults for Building State Files for import to the Radia Application Knowledge Base.

Using a Mouse to Manipulate Elements in the Window

You will probably be using a pointing device such as a mouse to work with objects in the Radia System Explorer. With its familiar graphical user interface, the Radia System Explorer supports standard mouse actions activated by clicking or double-clicking the left mouse button. Radia System Explorer also provides context sensitive shortcut menus activated by clicking the right mouse button.

The following table provides a summary of mouse actions you can perform.

Table 2.1 ~ Summary of Mouse Actions	
Mouse Action	Result
Click the left mouse button. (We refer to this as a left-click, or simply a click.)	Highlight an icon, select a drop-down menu command, or press a button.
Double-click the item using the left mouse button. (We refer to this as choosing an item.)	Open the next level window or dialog box, depending on the item to which the mouse is pointed.
Click the right mouse button. (We refer to this as a right-click.)	Display a shortcut menu containing choices appropriate to the item right-clicked.
Point.	Move the mouse so that the pointer displayed on your screen rests on the desired item.
Use the drag-and-drop operation by clicking on an object with the left mouse button, holding the mouse button down while moving the mouse pointer to the desired target, then releasing the mouse button.	Create a connection between two instances of connectable classes, or initiate a Notify action.

When using the Radia System Explorer to configure your enterprise desktop environment, you will use many familiar desktop elements: windows, drop-down menus, shortcut menus, menu commands, dialog boxes, and so forth.

Opening the Radia System Explorer

To open the Radia System Explorer

1. Click **Start**, **Programs**, and from the **Radia Administrator Workstation** menu choose **Radia System Explorer**.

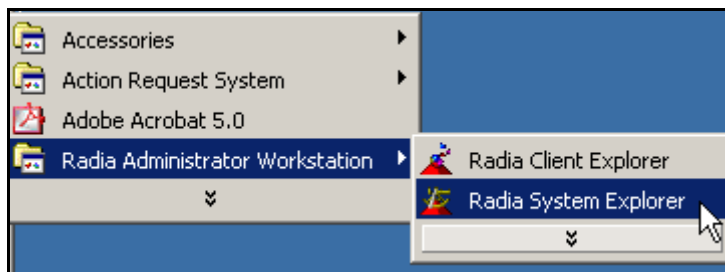


Figure 2.1 ~ Opening the Radia System Explorer.

A logon dialog box opens.

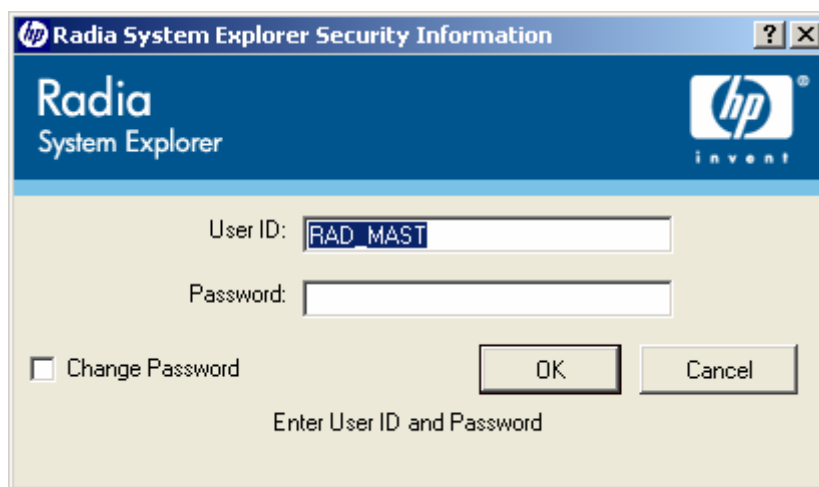


Figure 2.2 ~ The Radia System Explorer Security Information dialog box.

Note

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.

2. Click **OK**. The **Radia System Explorer** window opens.

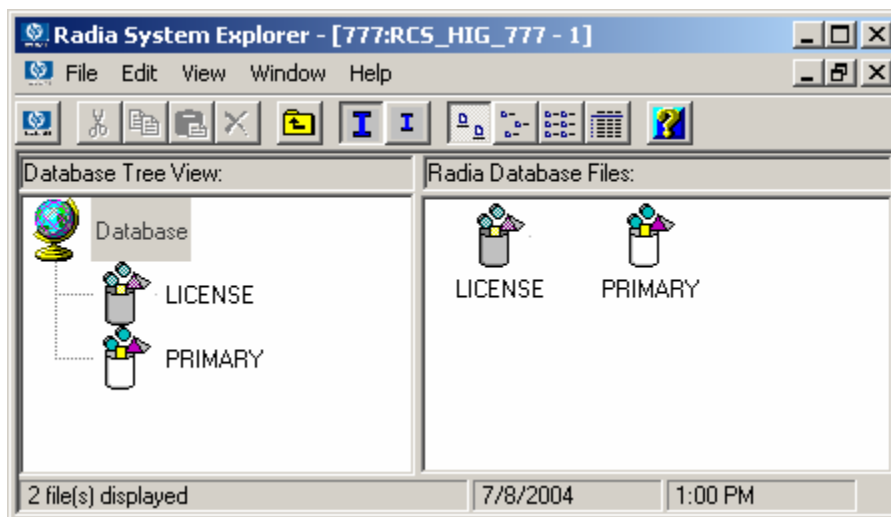


Figure 2.3 ~ Radia System Explorer window.

The Radia System Explorer has a familiar appearance, using a multiple document interface (i.e., you can open multiple Radia System Explorer windows simultaneously). Beneath the title bar, there is a drop-down menu, a tool bar, and a workspace containing Radia System Explorer windows. Each Radia System Explorer window contains a hierarchical tree view of the Radia Database on the left, and, on the right, a view of the contents of the Radia Database component currently selected in the tree view. A status bar opens at the bottom of the window.

When you first start the Radia System Explorer, the tree view displays a top-level view of the Radia Database, listing the files that comprise the Radia Database. You can expand and collapse the tree view to show or hide lower level components of the Radia Database, as described in *Navigating through the Radia Database* on page 46. With the Database icon selected in the tree view, the list view of the Radia System Explorer window displays a list of the files comprising the Radia Database.

The Radia System Explorer has many user-friendly features with which users of the Windows Explorer are familiar. For example, a number of items in the window provide tool tips. A tool tip is helpful information that opens spontaneously when the mouse pointer is moved over such an item in the Radia System Explorer window.

The Radia System Explorer Window

You can manipulate the appearance of the window and its contents by using customary Windows operations.

Basic Windows Operations

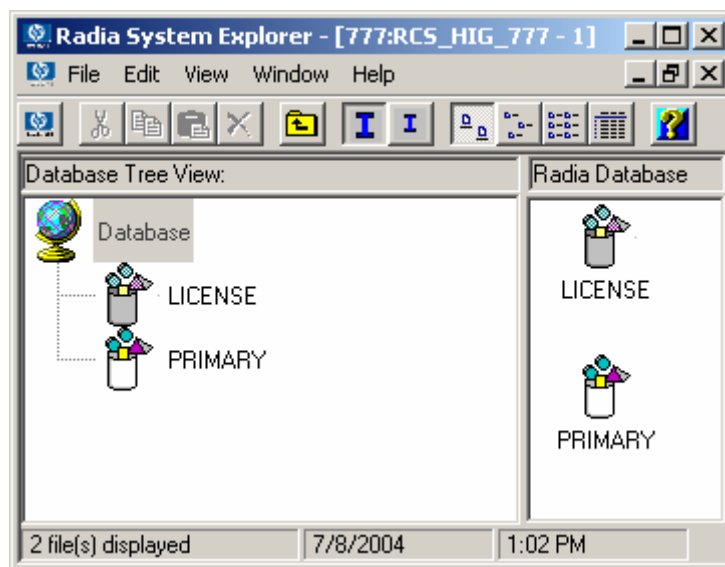


Figure 2.4 ~ You can manipulate the view of the window by clicking and dragging borders.

- Re-size the window by clicking and dragging the window borders or corners.
- Re-size the relative width of the tree view and the list view of the Radia System Explorer window by clicking and dragging the frame between them.

Customizing the Radia System Explorer Window

You can manipulate and customize the way the Radia System Explorer is displayed and how it functions by utilizing options from the various shortcut menus. The most common and frequently used functions are available from the **Radia System Explorer Toolbar**. See *The Radia System Explorer Toolbar* on page 31 for additional information about these functions.

Radia System Explorer Window Menus

■ The File Menu

Use the options in this window to open additional Radia System Explorer windows or to close the current window. Multiple windows can be open at the same time. You can drag items between multiple windows.

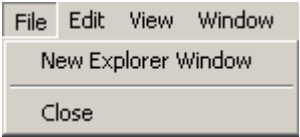


Figure 2.5 ~ The File menu.

■ The Edit Menu

Use the options in this shortcut menu to edit the Radia Database. The first six choices apply to the selected items in the active pane or dialog box.

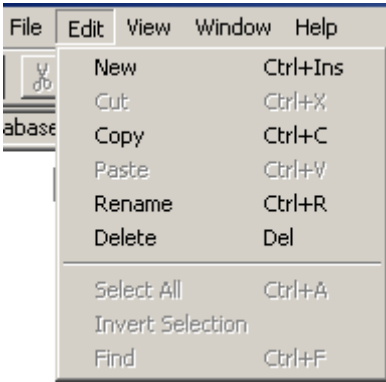


Figure 2.6 ~ The Edit menu.

Table 2.2 ~ Edit Menu

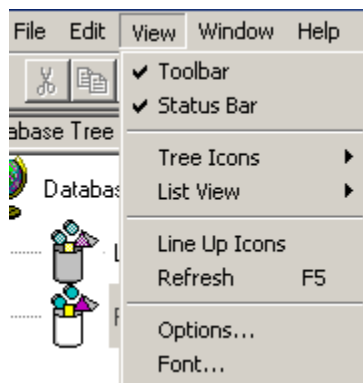
Operation	Description
New	Creates a new occurrence of the selected item such as domain, class, or instance.
Cut	Removes the selected text and places a copy of it on the Windows clipboard.

Table 2.2 ~ Edit Menu

Operation	Description
Copy	Replicates the selected text, or places a copy of it on 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 currently selected items from the Radia Database.
Select All	Selects all of the instances in the list view of an Radia System Explorer window.
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

The options from this shortcut menu control the appearance of the Radia Database. 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.

*Figure 2.7 ~ The View menu.***Table 2.3 ~ View Menu**

Option	Description
Toolbar	Click Toolbar to toggle the toolbar display on and off.
Status Bar	Click Status Bar to toggle the Status Bar display on and off.

Table 2.3 ~ View Menu

Option	Description
Tree Icons	Toggles the display between large and small icons.
List View	Choose from among four different displays for your list view (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 Radia System Explorer window, lining up all of the elements for easy reading.
Options	Opens the Radia System Explorer Options dialog box. See <i>Setting Preferences: The Options Dialog Box</i> on page 32 for more information.
Font	Opens a dialog box that allows you to change the font used to display text in the Radia System Explorer windows.

■ The Window Menu

Use the options in this window when you have more than one Radia System Explorer window open.

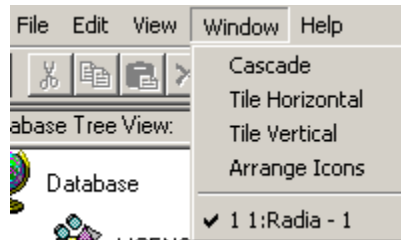


Figure 2.8 ~ The Window menu.

Table 2.4 ~ Window Menu

Item	Description
Cascade	Organizes and overlaps the windows starting at the top, left corner of the Radia System Explorer interface.
Tile Horizontal	Automatically sizes open windows to the width of the Radia System Explorer user interface. It allocates the available height within the Radia System Explorer interface equally, stacking the windows one on top of the other.
Tile Vertical	Automatically sizes open windows to the height of the Radia System Explorer interface. It allocates the available width equally, stacking the windows next to each other.
Arrange Icons	Docks minimized Radia System Explorer windows in the bottom left corner of the interface.

Table 2.4 ~ Window Menu

Item	Description
1 1:Radia – 1	Multiple View—allows the administrator to toggle between open Radia System Explorer sessions. A check next to a session indicates which window is currently active.

■ The Help Menu

Get details about the Radia System Explorer or link to the HP OpenView web site.

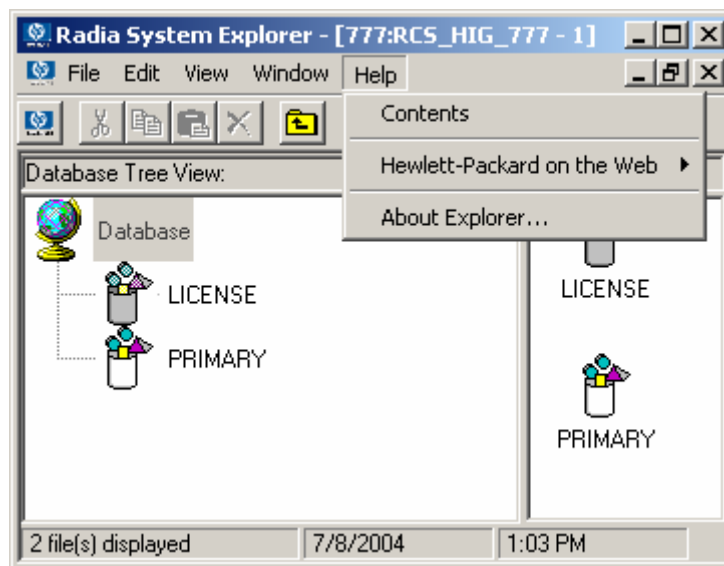


Figure 2.9 ~ The Help menu.

Table 2.5 ~ Help Menu

Item	Description
Contents	Opens a traditional Windows-style help interface. See Error! Reference source not found. starting Error! Bookmark not defined.
Search For Help On	This option has the same functionality and appearance as the Contents feature.
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.

Table 2.5 ~ Help Menu

Item	Description
About Explorer	Displays information about the current Radia System Explorer—Version number, Radia Configuration Server (Manager) Information and so forth, can be found here. 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 Radia System Explorer Toolbar














The Radia System Explorer toolbar provides easy access to frequently used functions:



Figure 2.10 ~ The Radia System Explorer toolbar.

The following table describes the toolbar buttons and their functions.

Table 2.6 ~ Radia System Explorer Toolbar

Icon	Action
	Open a new Radia System Explorer 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. Provides easy selection from a small list.
	Display small icons in the tree view. Maximizes the number of items that can be seen in the list view of the Radia System Explorer at one time. Provides easy selection from larger lists.
	Display large icons in the list view. Provides easy selection from a small list.
	Display small icons in the list view. Maximizes the number of items that can be seen in the list view of the Radia System Explorer window at one time. It is useful for selecting from large lists.
	Display the list view vertically. Maximizes the number of items that can be seen in the list view of the Radia System Explorer window at one time.
	Display the list view horizontally. Adds a column for TYPE (such as Class, Instance, and so forth). Maximizes the number of items and provides detailed description of the list view.
	Get context sensitive help. The cursor will change shape to indicate you are seeking help. Click this cursor on a control or screen area for which seeking help.

Setting Preferences: The Options Dialog Box

The Radia System Explorer provides an **Options** dialog box to allow you to control some of the features of the Radia System Explorer.

1. To open the **Options** dialog box, click **Options** from the **View** menu. The following dialog box opens.

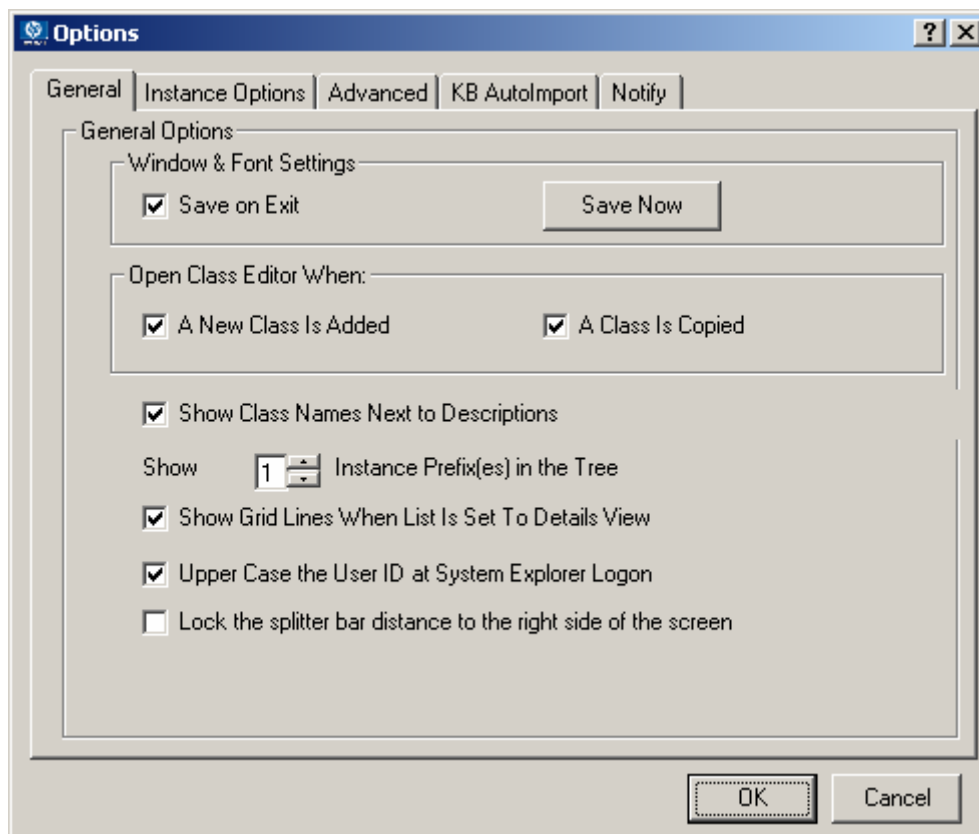


Figure 2.11 ~ The Options dialog box of Radia System Explorer with the General tab displayed.

There are five tabs in the **Options** dialog box:

- **General**
- **Instance Options**
- **Advanced**
- **KBAutoImport**
- **Notify**

2. To display the settings associated with a tab, click the tab.

General Tab

Table 2.7 ~ Settings on the General Tab and their Functions

General Tab	What It Does and How to Use It
Window & Font Settings	<p>Select Save on Exit to have the Radia System Explorer track certain window settings and the font to use for displaying text when you exit from the Radia System Explorer session.</p> <p>The window settings that are retained include the view setting (i.e., large icons, small icons, list, or details) for the tree and list views of the Radia System Explorer window, and the relative width of each pane. Window settings are saved independently for each level of the Radia Database.</p> <p>Click Save Now to have Radia System Explorer save the current window settings and font in use.</p> <p>You can change the font Radia System Explorer uses to display text by choosing a font from the dialog box that opens when you click Fonts from the View menu.</p>
Open Class Editor When	<ul style="list-style-type: none"> • Select the A New Class is Added check box to have Radia System Explorer open the Editing Class dialog box when you add a class. • Select the A Class is Copied check box to have Radia System Explorer open the Editing Class dialog box when you copy a class. <p>You will probably want to edit the newly copied or newly added class, so HP recommends that you have these two check boxes selected.</p>
Show Class Names Next to Descriptions	<p>Select this check box to show the Radia internal name for classes next to their friendly names in the tree view and in the list view of the Radia System Explorer window. The internal name will appear in parentheses next to the friendly name.</p> <p>This is useful if you ordinarily do not use the Details View for the list view of the Radia System Explorer window. The Details View in the list view of the Radia System Explorer window always shows the internal name for classes in the Type column.</p>
Show n Instance Prefix(es) in the Tree	<p>This option can be set to 0, 1, 2, or 3. It controls 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 <i>Navigating through the Radia Database</i> on page 46.</p>
Show Grid Lines When List Is Set To Details View	<p>Select this check box to have Radia System Explorer display grid lines in the list view when its view option is set to "Details". Clearing this check box removes the grid lines. Which setting results in a more legible display is a matter of personal preference.</p>
Uppercase the User ID at Radia System Explorer Logon	<p>Select this check box to automatically convert the user ID into all uppercase.</p>
Lock the splitter bar distance to the right side of the screen	<p>Select this check box to lock the width of the right side of the Radia System Explorer window during resizing. This is useful if you are focusing on information on the right side of the Radia System Explorer Window. The left side will increase or decrease in size as you resize the window.</p> <p>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.</p>

Instance Options Tab

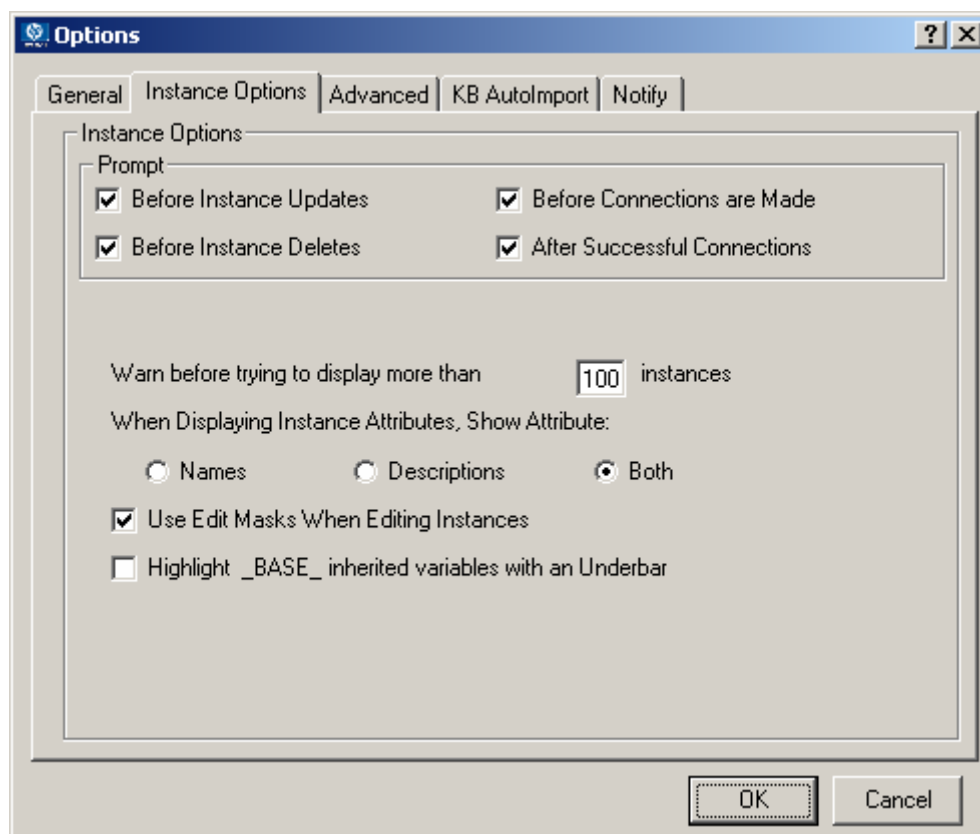


Figure 2.12 ~ The Instance Options tab of the Options dialog box.

The following settings are available on the **Instance Options** tab.

Table 2.8 ~ Settings on the Instance Options Tab

Instance Options Tab	What It Does and How To Use It
Prompt	<p>These four check boxes control whether Radia System Explorer prompts you before completing the actions indicated by their descriptions. The prompt consists of a message asking you to confirm your desire to complete the action before anything is permanently changed in the Radia Database. Selecting these is a matter of personal preference.</p>
Warn before trying to display more than n instances	<p>In large distribution models, 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 similar to the following opens:</p> <div data-bbox="736 484 1028 680" data-label="Image"> </div> <p>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 Radia System Explorer retrieves them from the Radia Database, and you will be inconvenienced by the need 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 Radia System Explorer 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.</p> <p>If this check box is selected, attributes that hold logical (i.e., Yes/No) data have check boxes in the data entry area. Attributes that have designated 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.</p> <p>You can set the desired 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 this check box is not selected, all data entry in the Edit Instance dialog box will be plain text.</p> <p>This facility is described in detail in <i>Optional Attribute Editing Controls</i> on page 67.</p>
Highlight _BASE_ inherited variables with an Underbar	<p>Use this setting if you want to be able to see which values in an instance are inherited from the <code>_BASE_INSTANCE_</code> of the class. The inherited instances will show with a red underbar below them.</p>



Advanced Tab

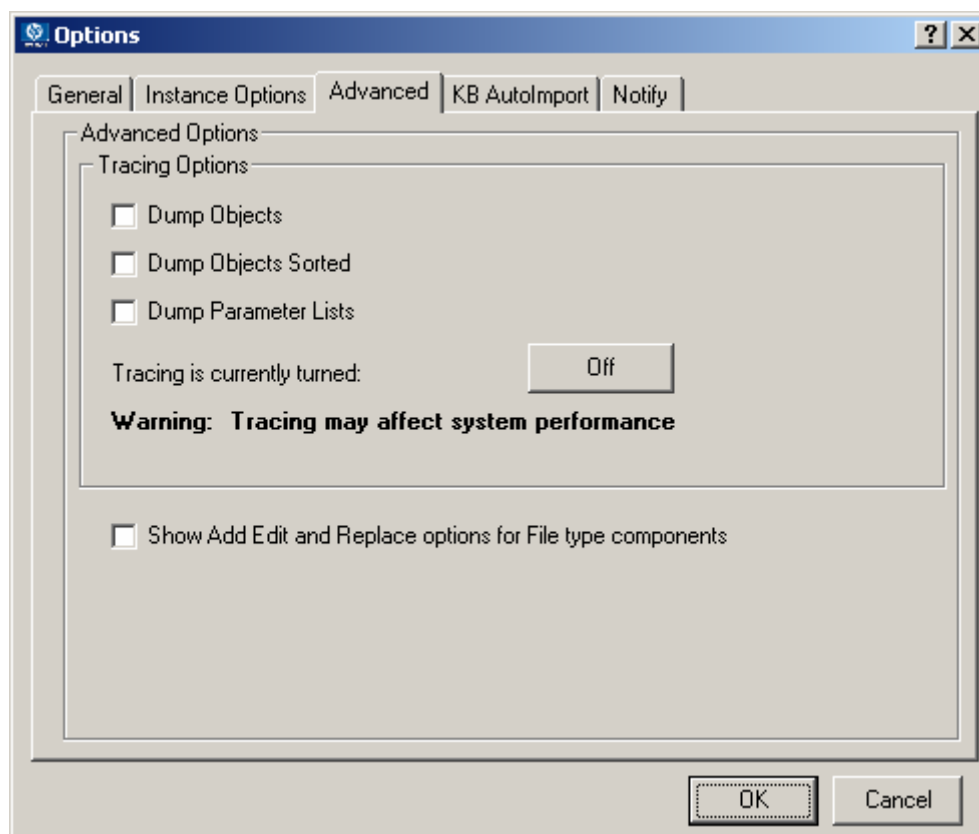


Figure 2.13 ~ The Advanced tab of the Options dialog box.

Table 2.9 ~ Advanced Options Tab

Advanced Tab	What It Does and How To Use It
Tracing Options	These settings are only used for troubleshooting the Radia System Explorer. Do not change these settings without consulting HP Technical Support.

Table 2.9 ~ Advanced Options Tab	
Advanced Tab	What It Does and How To Use It
Show Add, Edit and Replace options for File type components	<p>WARNING: This setting must only be used in non-production environments.</p> <p>This setting determines whether or not to display the following commands on the shortcut menus for file-type components:</p> <ul style="list-style-type: none">• Edit this Component• Replace Component Data• Add a Component <p>For details, see <i>Appendix C: Modifying File-type Components and Data (Non-Production Only)</i>.</p>

KB AutoImport Tab

Use the **KB AutoImport** tab to define the Radia Knowledge Base Manager default auto-import directory. The Radia Knowledge Base Manager is a component of the Radia Extension for Windows Installer. For details, see the *Radia Extension for Windows Installer Guide*.

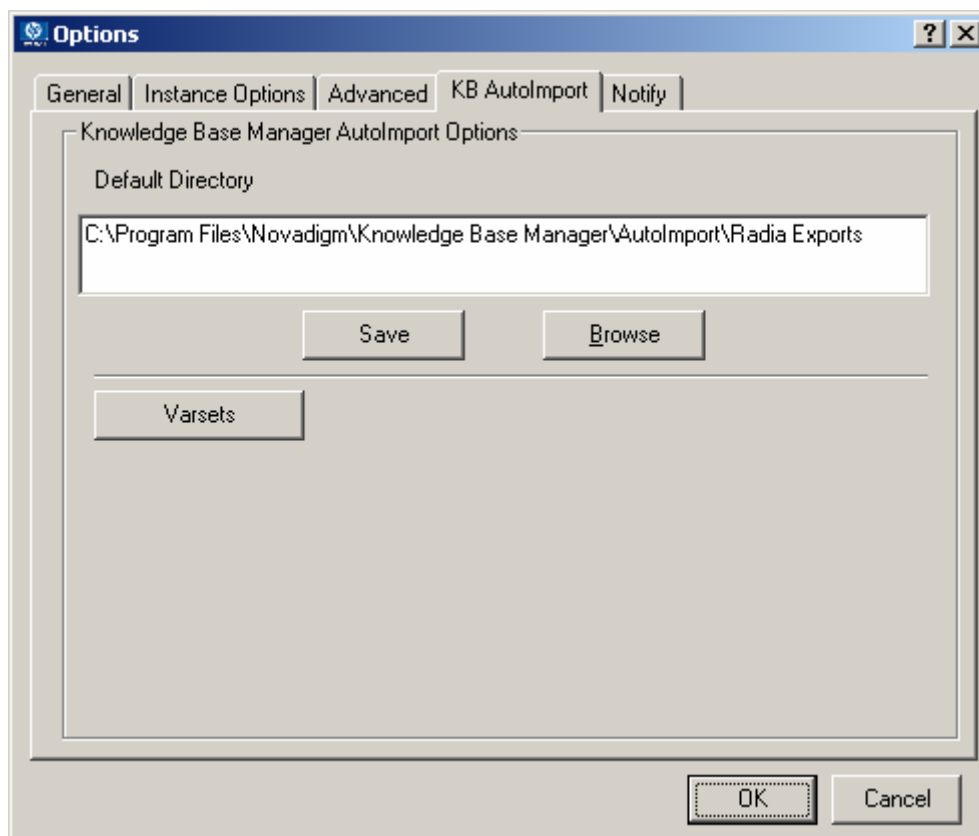


Figure 2.14 ~ The KB AutoImport tab of the Options dialog box.

To specify the Knowledge Base Manager AutoImport options

1. Specify or browse to the **Default Directory** to be used as an export directory when you Build State Files for import to the Radia Knowledge Base. The default location is the **\Radia Exports** directory installed by the Radia Knowledge Base Manager. This directory includes the required variable set subdirectory: **\VarSets**.

Caution

Any directory that will be used as an export directory must include a subdirectory named **\VarSets**.

2. Click **Save** on the KB AutoImport tab to ensure the default directory you specified for the Knowledge Base Manager default auto-import directory is set.

For details on building state files from application services, see **Error! Reference source not found.** on page **Error! Bookmark not defined.**

Notify Tab

Use the **Notify** tab to specify the radskman.exe command line values for UID and STARDIR that will be sent to clients when they are notified. (The UID is the identification 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 clients by machine name, as opposed to the name of the user currently logged on.

The Notify options set here are used by the Radia Notification Manager and displayed on the Notifications Detail dialog box. See the topics *Radia Notify* on page 174, *Notify Dialog Boxes* on page 176, and *Set Notification Details* on page 180 for more information.

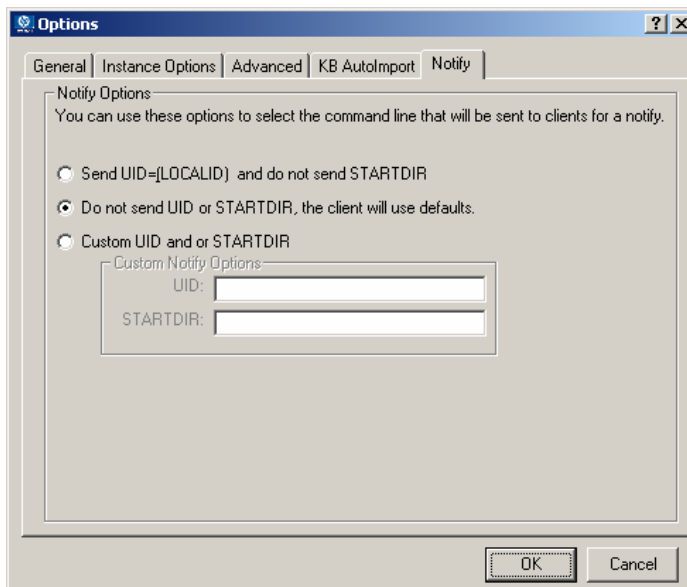


Figure 2.15 ~ The Notify tab of the Options dialog box

Table 2.10 ~ Settings on the Notify Options Tab

Notify Option	What It Does and How To Use It
Send UID=[LOCALID] and do not send STARTDIR	Creates a Notify command with UID=[LOCALID]. This command Notifies clients by the name of the user currently logged on. To Notify clients by the name of the machine, use the Custom UID or STARTDIR option, below.
Do not send UID or STARDIR, the client will use defaults.	Creates a Notify command without any values for UID or STARTDIR. Clients use default values for UID and STARTDIR. To specify non-default values, use the Custom UID and STARTDIR option, below.
Custom UID 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 clients by the 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 clients 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.

Summary

- Use a pointing device, such as a mouse, to work with objects in the Radia System Explorer.
- You can manipulate the appearance of the Radia System Explorer window and its contents by using customary Windows operations.
- Frequently used functions of the Radia System Explorer window are available from the Radia System Explorer Toolbar.
- The Radia System Explorer provides the administrator with the ability to control some of the features and options of the Radia System Explorer.

Radia Database Overview

At the end of this chapter, you will:

- Understand the schema of the Radia Database and how to navigate through it.
- Understand default values for instances and how inheritance works.
- Be familiar with the various Radia System Explorer options available through the shortcut menus.
- Know how to work with Radia domains, classes, and instances.

The Radia Database

The Radia Database, located on the Radia Configuration Server, stores your distribution model. This includes all of the information that Radia uses to manage applications on a client computer, including:

- The software or data that Radia distributes.
- The distribution model for each client computer.
- The policies determining which subscribers are assigned to which packages.
- Security and access rules for Radia administrators.

Use the Radia System Explorer to view and manipulate the Radia Database.

When you install the Radia Configuration Server, **LICENSE** and **PRIMARY** are the only two files available. As you use Radia, your Radia Database will change.

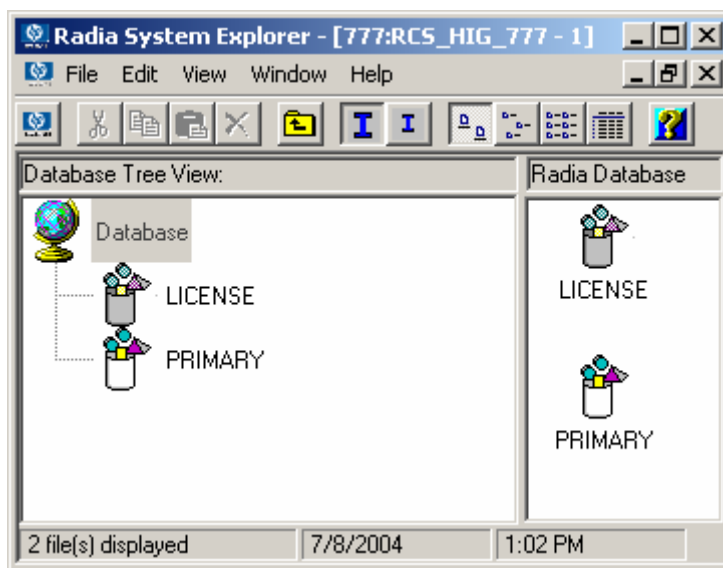


Figure 3.1 ~ The Radia Database—*LICENSE* and *PRIMARY* files.

- The **LICENSE** file is read-only and used for Radia Configuration Server processing. This file is for HP use and should not be modified.

- The **PRIMARY** file is where you will find most of the information regarding software management. Within the PRIMARY file, there are seven default domains.
 - Use the **ADMIN** domain to limit administrative access to the Radia Database, as well as specify rules for connecting classes.
 - Use the **AUDIT** domain to configure tasks that will inventory client computers' assets. See the *Radia Inventory Manager Guide* for additional information.
 - Use the **NOVADIGM** domain to store packages for self-maintenance. See *Radia Software Manager Self Maintenance* in the *Publishing Applications and Content* chapter in the *Radia Software Manager Guide* for more information.
 - Use the **PATCH** domain to store the list of available patches within a PATCH class. See *Service Optimization* on page 145 for additional information about this domain.
 - Use the **POLICY** domain to organize subscribers into logical groups. Each client computer managed by Radia is represented by an instance of the POLICY domain USER class. See the *Implementing Entitlement Policy* chapter in the *Radia Software Manager Guide* for more information.
 - Use the **SOFTWARE** domain to store information about the software being managed and the methods used to deploy the software. Each ZSERVICE instance represents one software application or a single unit of other content to be managed on client computers.
 - Use the **SYSTEM** domain to store administrative and process control definitions.
- As you begin to use Radia, the **PROFILE** and the **NOTIFY** files appear.
 - The **PROFILE** file appears after the first client computer has registered with the Radia Configuration Server. The PROFILE file contains information collected from client computers. This information is used to connect to computers to deploy software managed by Radia as well as to see the configuration of the client computer.
 - The **NOTIFY** file appears once a notify has been initiated. The NOTIFY file is divided into domains, where each domain is one Notify operation. Each Notify file domain has one NOTIFY class. Each NOTIFY class contains an instance for each selected member of the Notify operation's audience list. For additional information, see *Radia Notify* on page 174.

Navigating through the Radia Database

The Radia Database 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 Radia Database components.

To navigate through the tree structure

1. Open the Radia System Explorer, and double-click the PRIMARY file in the tree view.

The **Radia System Explorer** window will be similar to the following:

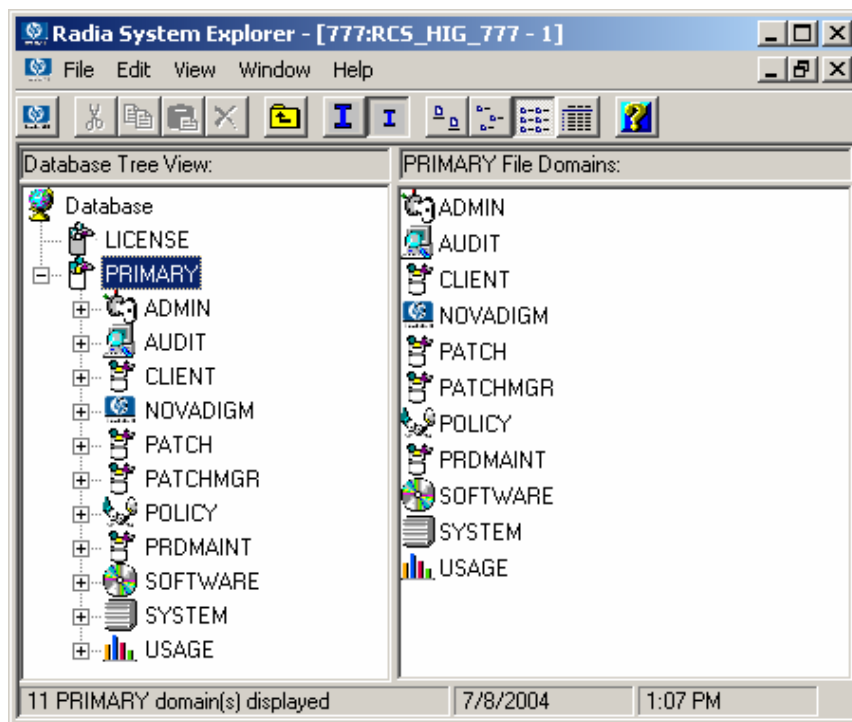


Figure 3.2 ~ Navigating through the tree structure.

The seven domains of the PRIMARY file appear as branches beneath its icon in the tree view, and simultaneously appear as individual icons in the list view of the Radia System Explorer window. A small box with a minus sign appears in the tree structure next to the PRIMARY file icon, signifying that the tree is displaying the next level of the Radia Database hierarchy

below the PRIMARY file. You can click the minus sign to close the tree view below that level. When you do so, the tree closes and the minus sign changes to a plus sign.

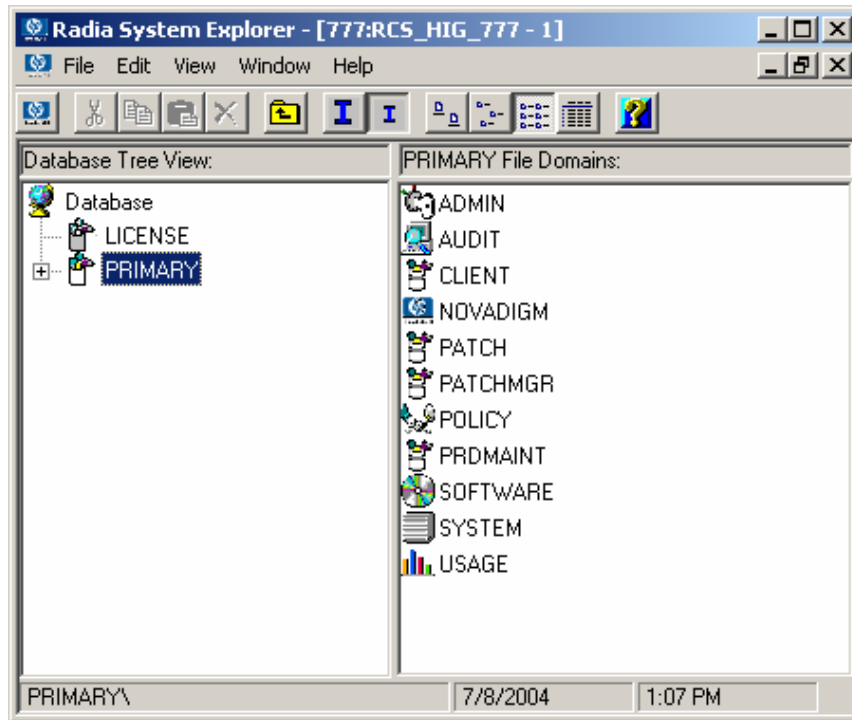


Figure 3.3 ~ Clicking the minus sign closes the tree view below that level.

2. Click the plus sign, and the previous tree view will be restored.
3. To open a Radia Database component, double-click its icon.

For example, to display a component from the next level down the hierarchy, double-click the SOFTWARE domain icon either in the tree view or on the list view of the Radia System Explorer window. The classes of the SOFTWARE domain will be displayed in the list view of the Radia System Explorer window.

If you double-click the SOFTWARE domain icon in the tree view, the tree view opens to display the SOFTWARE domain classes below its icon. If you double-click the SOFTWARE domain icon on the list view of the Radia System Explorer window, a small box containing a plus sign appears next to the icon in the tree view, but the classes are not displayed beneath it.

Navigation History

As you open and close components in the Radia System Explorer, the tree view keeps track of which branches you've opened and closed. It marks previously opened and closed branches at each non-elementary level with a plus or minus sign in a small box next to the component's icon. Clicking the plus sign expands the branch of the tree view to the highest level of detail previously opened in the current Radia System Explorer session. Clicking the minus sign collapses the tree view below the clicked node. 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.

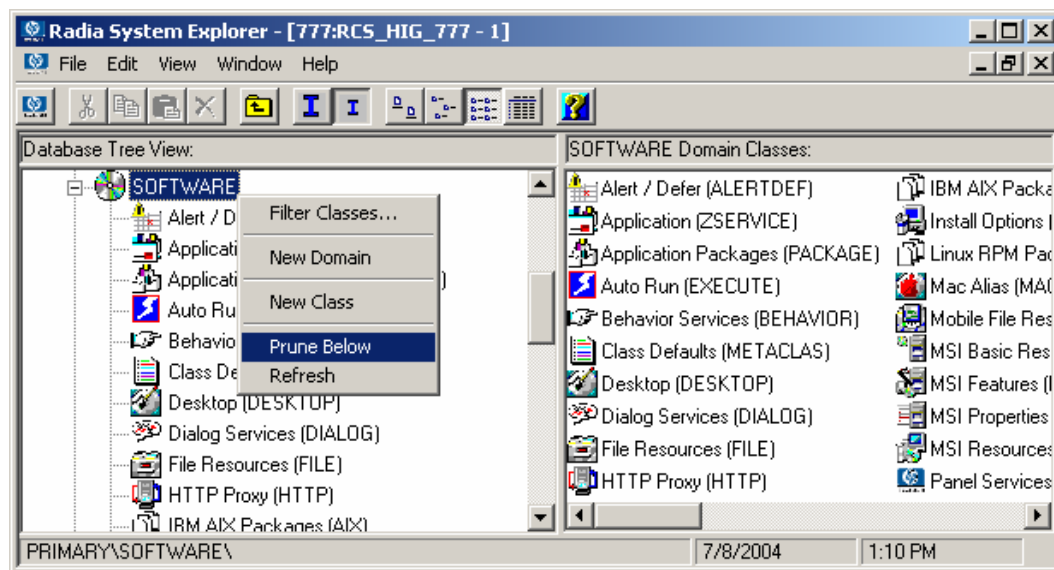


Figure 3.4 ~ Use the Prune Below function to discard the history of previously opened branches beneath a component.

2. Click **Prune Below**.

The tree view collapses all open nodes below the SOFTWARE domain, and discards the history of their having been open.

To rebuild the display of a node and its branches in the tree view

From the current data in the Radia Database, use the **Refresh** function.

1. Right-click the SOFTWARE domain icon in the tree view.

A shortcut menu opens.

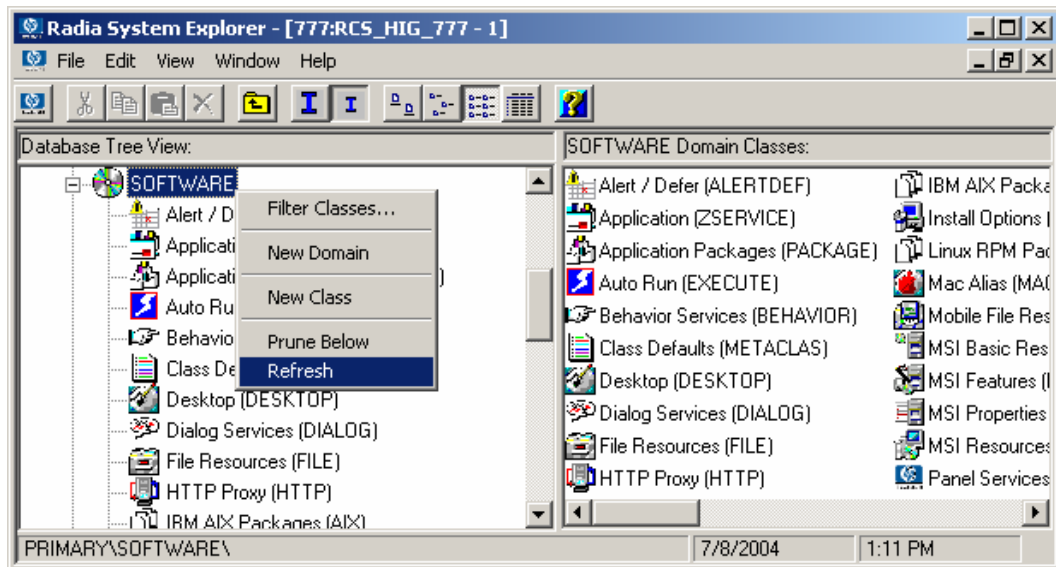


Figure 3.5 ~ Use the Refresh function to rebuild the display of a node and its branches in the tree view.

2. Click **Refresh**.
3. The tree view collapses all open nodes below the SOFTWARE domain, retrieves all data for the SOFTWARE domain from the Radia Database, and re-expands the tree view of the SOFTWARE domain to show the classes it contains. The display now includes all changes made to the Radia Database during the current session.

When you **Refresh** a node in the tree view, the data for that node (and its sub-nodes) will be retrieved from the Radia Database, and the refreshed node will be expanded in the tree view.

Contrast this with the **Prune Below** operation and expanding/collapsing sub-nodes by clicking the plus or minus sign next to a node. Neither of these retrieves data from the Radia Database; they simply change the view of data already retrieved.

Compound Names

As you configure more and more desktops with increasingly more applications, the Radia Database can grow quite large. For example, you will have an instance of the USER class for each desktop being managed and an instance of the FILE class for each file to be distributed to a desktop. Depending on the size of your enterprise this can entail many thousands of instances.

Manipulating the tree view can grow unwieldy when large numbers of instances are involved. Radia helps you manage large numbers of instances by providing a filtering capability (described in *Filtering—Viewing Portions of the Radia Database* on page 187). Radia also provides one, two, or three additional collapsible levels in the tree view, based upon one, two, or three high-level qualifiers of the instance name.

To take advantage of the additional collapsible levels in the tree view, you must name your instances consistently with compound names. 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.

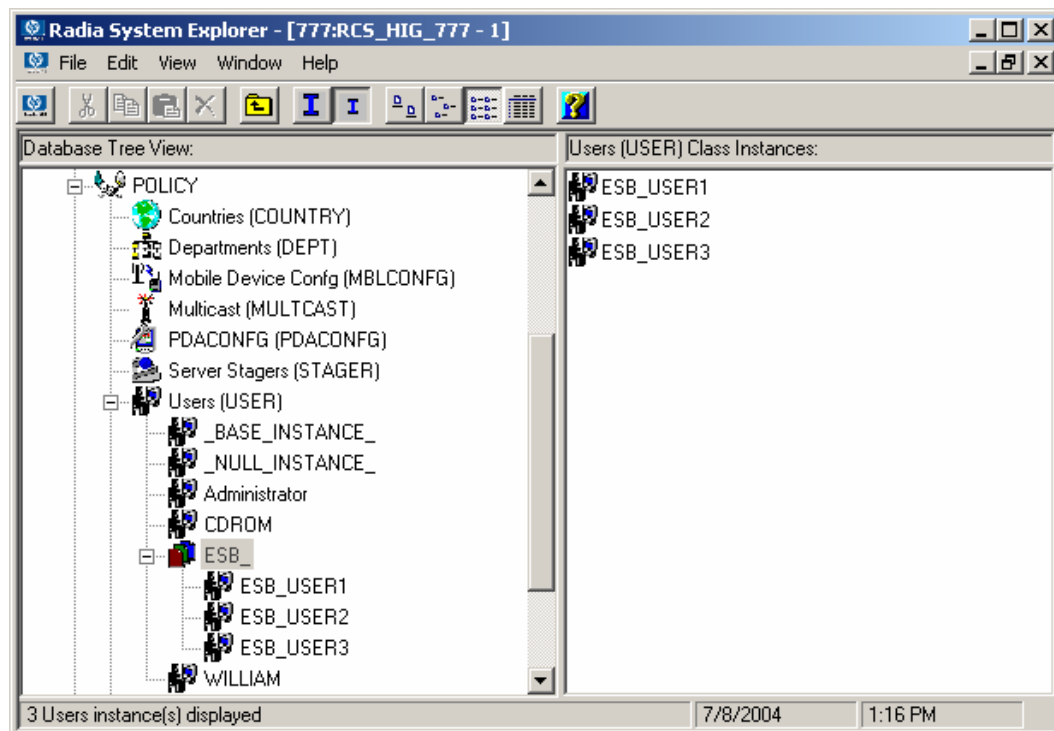


Figure 3.6 ~ Compound names are used to manage large numbers of instances.

Like any other non-elementary level, clicking on the minus sign at the next higher level collapses this level of the tree view.

The number of prefix levels in an instance name (none, one, two, or three) 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 32.

Each underscore in an instance name delimits 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 Radia Database

Each class in the Radia Database has a base instance, named `_BASE_INSTANCE_`.

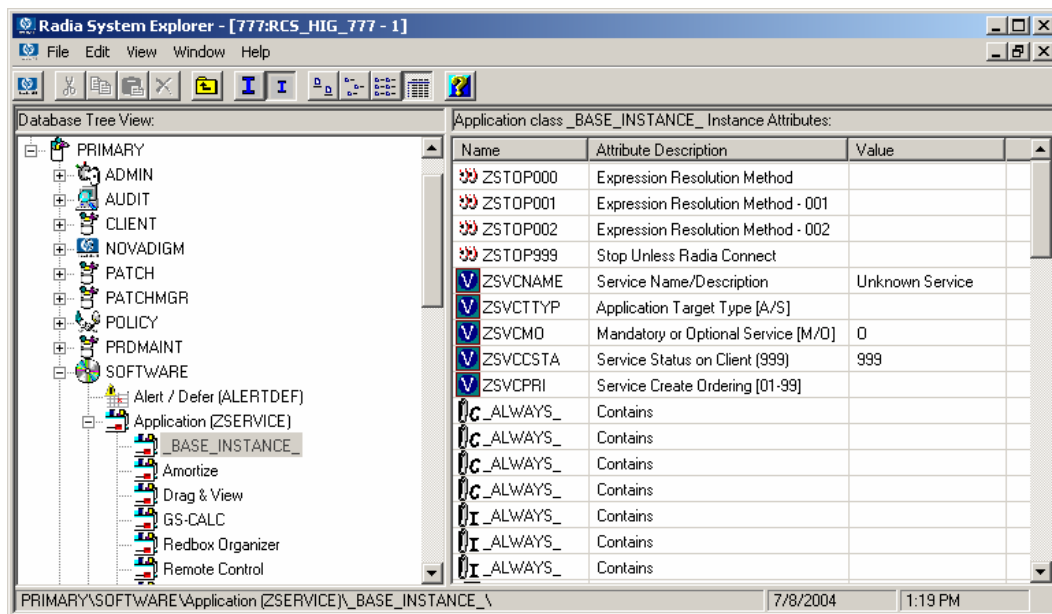


Figure 3.7 ~ An example of a `_BASE_INSTANCE_` in the Application class.

The purpose of the base instance is to provide default values for other instances of the class. Any attribute in an instance will assume the value for that attribute found in the base instance of the class, unless a specific value has been entered for that attribute in the instance.

Inherited values appear in the instance no differently from 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 55 to learn how), and the value re-appears in the instance automatically, the value is being inherited from the base instance.

Radia System Explorer Shortcut Menus

In the Radia System Explorer you can add, copy, edit, rename, and delete Radia Database components (domains, classes, and instances). It also provides extensive support for making correct connections between instances.

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

For a complete list of the shortcut menus, see *Radia System Explorer Shortcut Menus* starting on page 267.

Working with Domains

Adding a Domain

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 following dialog box opens.

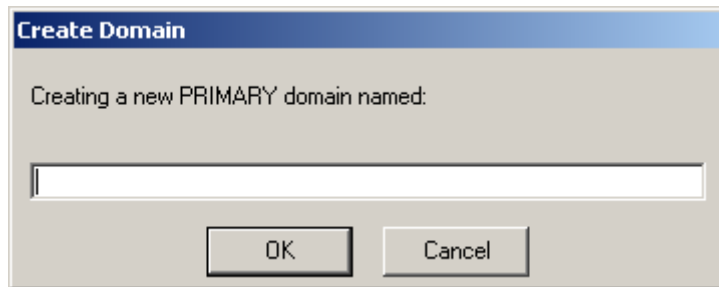


Figure 3.8 ~ Create Domain dialog box.

2. Enter a name for the new domain and click **OK**.

Domain names can be a maximum of eight characters including letters and numbers.

Working with Classes

Adding a Class

New classes can be added by

- copying an existing class,
- OR
- adding a new class from scratch.

Which to choose depends upon whether a class exists whose attributes are substantially similar to the class you want to add. If so, copy the existing class; otherwise, add a new class from scratch.

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.

Copying a Class

To copy an existing class

1. Highlight the class to be copied.
2. Right-click on the class.
3. Select **Copy Class** from the shortcut menu.

The following dialog box opens.

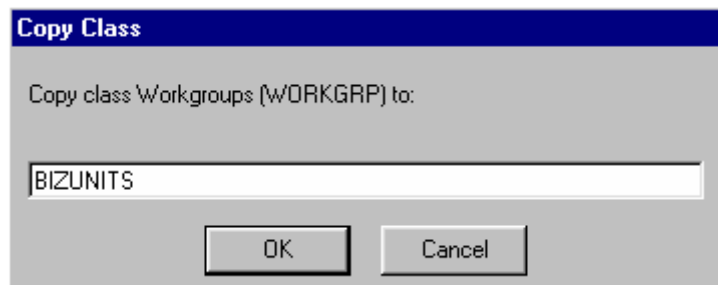


Figure 3.9 ~ Copy Class dialog box.

4. Enter a name for the new class (BIZUNITS in this example), and click **OK**.
 - If the option to open the Class Editor when a class is copied is in effect (see *Setting Preferences: The Options Dialog Box* on page 32 for details), the following **Editing Class** dialog box opens. See *Editing a Class* on page 61.

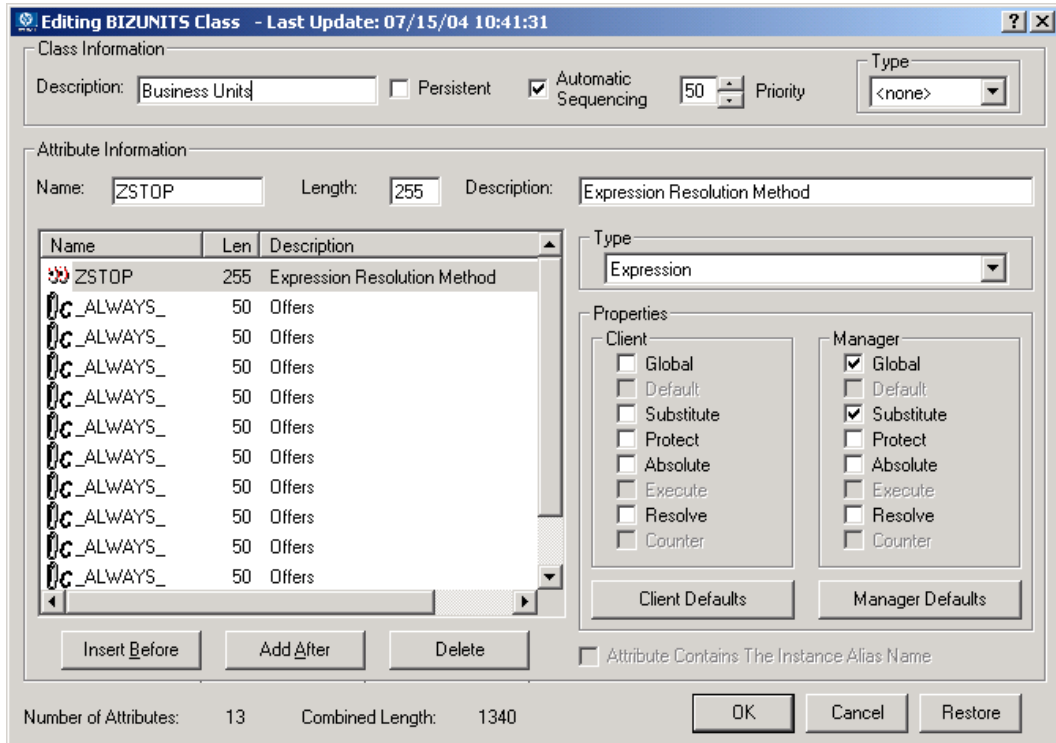


Figure 3.10 ~ The Class Editing dialog box opens when the option Open Class Editor When a Class is Copied is active.

- If the option to open the Class Editor when a class is copied is not in effect, the original class is copied with the new name, the Class Editor is not opened, and no warning (see next step) is issued. You will have to manually open the Class Editor to edit the newly copied class.
5. To complete the addition of the copied class, change the **Description** field to identify the newly added class, edit the class to reflect the new class' requirements, and click **OK**.

The following message opens.

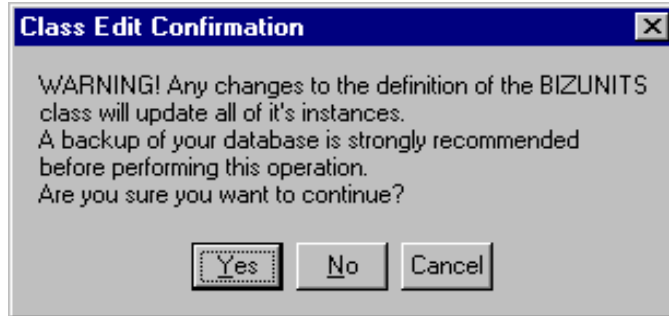


Figure 3.11 ~ Class Edit Confirmation dialog box.

6. Click **Yes**, and the new class addition is complete.

Note

Clicking **No** or **Cancel** to the message in Figure 3.11 does not terminate the process of copying the class. It merely closes the Class Editing dialog box. If you have created a class and wish to delete it, see *Deleting a Class* on page 72.

Once you have created your class, you will need to check the MASTER_CONNECTION_RULES variable to determine if you will 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 Radia Database* beginning on page 211.

Create a New Class from Scratch

To add a new class from scratch

1. Highlight the domain where you wish to insert the new class.
2. Right-click on the domain name.
3. Select **New Class** from the shortcut menu.

The following dialog box opens.

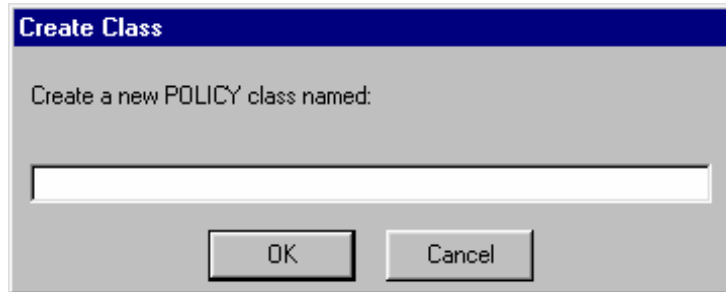


Figure 3.12 ~ Create Class dialog box.

4. Enter a name for the new class (BIZUNITS in this example), and click **OK**.
 - If the option to open the Class Editor when a new class is added is in effect (see *Setting Preferences: The Options Dialog Box* starting on page 32 for details), the **Editing Class** dialog box will open. The **Editing Class** dialog box is described in *Editing a Class* on page 61.

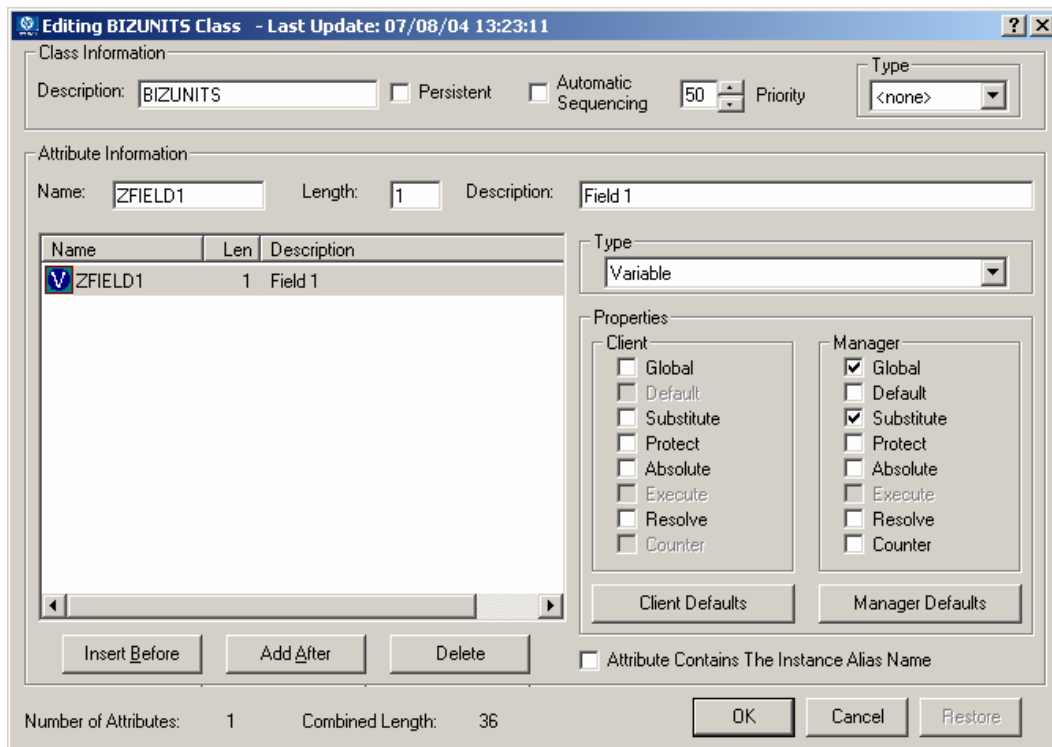


Figure 3.13 ~ Editing Class dialog box.

- If the option to open the Class Editor when a new class is added is not in effect, a new class is added with the specified name, the Class Editor is not opened, and no warning (see next step) is issued. You will have to manually open the Class Editor to edit the newly added class.
5. To complete the addition of the new class, change the **Description** field to identify the newly added class, edit the class to reflect the new class' requirements, and click **OK**.

The following message opens.

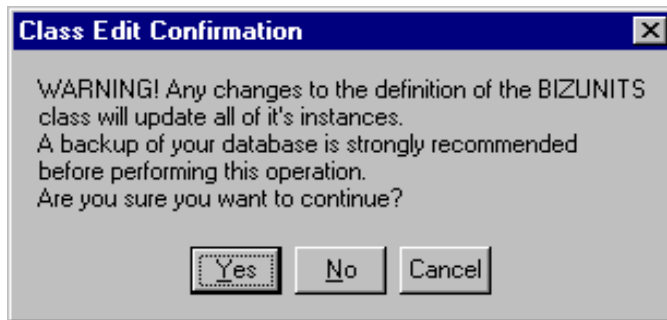


Figure 3.14 ~ Class Edit Confirmation dialog box.

6. Click **Yes**, and the new class addition is complete.

Note

Clicking **No** or **Cancel** to the message in Figure 3.14 does not terminate the process of creating the class. It merely closes the Class Editing dialog box. If you have created a class and wish to delete it, see *Deleting a Class* on page 72.

Once you have created your class, you will need to check the MASTER_CONNECTION_RULES variable to determine if you will 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 Radia Database* beginning on page 211.

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 55, the **Editing Class** dialog box opens.

To edit a class

1. Right-click on the Departments class in the tree view, and choose **Edit Class** from the resulting shortcut menu.

The **Editing Class** dialog box opens.

Editing DEPT Class - Last Update: 12/11/00 11:49:59

Class Information

Description: ☐ Persistent ☐ Automatic Sequencing Priority

Attribute Information

Name: Length: Description:

Name	Len	Description
<input checked="" type="checkbox"/> ACCTNO	12	Account Number
<input checked="" type="checkbox"/> _ALWAYS_	50	Offers
<input checked="" type="checkbox"/> _ALWAYS_	50	Offers
<input checked="" type="checkbox"/> _ALWAYS_	50	Offers
<input checked="" type="checkbox"/> _ALWAYS_	50	Offers
<input checked="" type="checkbox"/> _ALWAYS_	50	Offers
<input checked="" type="checkbox"/> _ALWAYS_	51	Utility Resolution Method
<input checked="" type="checkbox"/> NAME	80	Friendly name

Type:

Properties

Client

☐ Global ☐ Default ☐ Substitute ☐ Protect ☐ Absolute ☐ Execute ☐ Resolve ☐ Counter

Manager

☒ Global ☐ Default ☒ Substitute ☐ Protect ☐ Absolute ☐ Execute ☐ Resolve ☐ Counter

☐ Attribute Contains The Instance Alias Name

Number of Attributes: 9 Combined Length: 758

Figure 3.15 ~ The Editing Class dialog box.

2. Set the **Class Information** properties at the top of the dialog box as described in the following table.

Table 3.1 ~ Class Information Properties

Class Information Property	Significance and Setting																
Description	The friendly name that describes the class and its purpose. This name appears next to the class icon in the tree view.																
Persistent	<p>This property is available to Radia administrators with a thorough understanding of the Radia resolution process.</p> <p>A persistent class is one who's attributes and objects remain in global memory for the duration of the resolutions process. The instances created from persistent classes are available to the Radia Client for storage on its desktop.</p> <p>If a class is not persistent, it is transient. A transient object exists only while it is being process during resolution. Instances created from transient classes are not available to the Radia Client for storage on the desktop.</p> <p>For additional information regarding persistent and transient objects and how they are incorporated into the resolution process, refer to <i>Understanding the Resolution Process</i> starting on page 199.</p>																
Automatic sequencing	<p>Normally, resolution proceeds from attribute to attribute in the order the attributes appear in the Attribute List. Refer to <i>Understanding the Resolution Process</i> starting on page 199.</p> <p>Selecting this check box imposes a grouping on the order in which attributes of the class are processed during resolution:</p> <ul style="list-style-type: none"> • Expressions • Variables • Classes (connections) • Methods <p>Groups are processed in the order in which they appear above, and within these groupings, the attributes are processed in the order in which they appear in the Attribute List.</p> <p>Unless you need to alter the normal sequence of processing attributes during resolution, this check box should be cleared.</p>																
Priority	<p>This field is two characters long and has a value between 00 and 99. During object resolution, the Radia Configuration Server puts this value in the ZOPJPRI variable. If there is a blank or zero value in this field, the Radia Configuration Server supplies the default value specified for that particular class. The default values for the classes are as follows:</p> <table> <tr> <td>PATH</td><td>5</td></tr> <tr> <td>REGISTRY</td><td>60</td></tr> <tr> <td>DESKTOP</td><td>70</td></tr> <tr> <td>FILE</td><td>50</td></tr> <tr> <td>METACLAS</td><td>10</td></tr> <tr> <td>ZSERVICE</td><td>50</td></tr> <tr> <td>MACALIAS</td><td>70</td></tr> <tr> <td>All others</td><td>50</td></tr> </table>	PATH	5	REGISTRY	60	DESKTOP	70	FILE	50	METACLAS	10	ZSERVICE	50	MACALIAS	70	All others	50
PATH	5																
REGISTRY	60																
DESKTOP	70																
FILE	50																
METACLAS	10																
ZSERVICE	50																
MACALIAS	70																
All others	50																

Table 3.1 ~ Class Information Properties

Class Information Property	Significance and Setting
Type	<p>Classifies classes according to their purpose and usage. Radia uses this classification to affect the resolution process. For example, resolution to populate the Radia Service List on a client computer does not process classes classified as "Component" classes.</p> <p>The following values can be selected from the drop-down list:</p> <ul style="list-style-type: none"> • <none> – This class is not classified. • Configuration – This class identifies content to be managed by Radia on client 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, DESKTOP, and so forth. • Policy – This class identifies one, or a group of client 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** drop-down list indicates the type of the highlighted attribute.
4. Type your desired changes into the **Attribute Information** data entry fields, and click on the desired **Type** option to set the attribute's type. Class attributes can be one of four types as described in Table 3.2 below.

Table 3.2 ~ Class Attribute Types and their Functions






Attribute Type	Usage
Expression 	Contains a single line REXX command that is executed during resolution. In an attribute named ZSTOP, causes resolution of the current instance to terminate, if the expression evaluates to "true" .
Variable 	A piece of named storage containing a variable value. The variable's value forms a part of the client's resolved distribution model, and can influence the resolution process through messaging or symbolic substitution.

Table 3.2 ~ Class Attribute Types and their Functions

Attribute Type	Usage
<p>Connection (available)</p> 	<p>Class connections determine the path of resolution of a client's distribution model during the client connection process.</p> <p>Available connections are attributes into which a connection to another class may be set. Set connections are attributes containing a specification for a connection to another class.</p>
<p>Connection (set)</p>  <p>Note: The letter in the icon denotes the connection type:</p> <p>A – Always C – Class I – Includes R – Requires</p>	<p>A class connection is, in effect, a branch in the resolution process. Refer to <i>Understanding the Resolution Process</i> starting on page 199.</p> <ul style="list-style-type: none"> • An <i>Always connection</i> is always resolved, no matter what the name of its attribute. • A <i>Class connection</i> 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 <i>Includes connection</i> is always resolved, no matter what the name of its attribute. • A <i>Requires connection</i> 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><i>Requires</i> and <i>Includes</i> connection types are useful, among other uses, for expressing inter-package relationships, connecting instances of the PACKAGE class.</p> <p>When one package <i>Requires</i> another, components of the required package are superseded by the same component in the requiring package.</p> <p>When one package <i>Includes</i> 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.

Client Properties

Caution

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

- The Radia Client **Protect** property, when selected, indicates that the value of an attribute stored in an object on the client computer will be encrypted.
- The Radia Client **Execute** property, which applies only to method attributes, indicates, when selected, that the method to be executed will be executed on the client computer.

Manager Properties

The following table summarizes the significance of the Manager properties.

Table 3.3 ~ Manager Properties and their Functions

Property	Significance and Setting
Global	When selected for a variable in a transient class, 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 dissolved.
Default	Used internally by Radia and is not available to Radia administrators. When selected, prevents default values in transient-class objects from being incorporated into parent, persistent-class objects.
Substitute	<p>When selected, 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 from occurring.</p> <p>The Manager Substitute property determines whether symbolic substitution occurs when a variable is created as a result of a Radia 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 Manager Substitute check box was selected for that variable when the object was created, substitution occurs as the variable is created in storage.</p> <p>For example, the symbol: &(ZMASTER. ZUSERID)</p> <p>will evaluate to: WILLIAM</p> <p>after symbolic substitution, if the value of the ZUSERID attribute of the ZMASTER object is WILLIAM at the time symbolic substitution occurs.</p>
Protect	When selected, 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 Radia Database.
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	When selected, indicates that the value of the attribute identifies a method to be executed by the Radia Configuration Server.

Table 3.3 ~ Manager Properties and their Functions

Property	Significance and Setting
Resolve	<p>When selected, 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 from occurring.</p> <p>The Manager 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 Radia Database and instantiated as an object in storage.</p>
Counter	<p>When selected, this check box indicates that the variable contains an integer that the Radia Configuration Server will accumulate. The accumulated total will be stored in a variable with the same name in all parent persistent objects.</p> <p>For example, this property can be used for such tasks as 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 Radia Configuration Server when the Manager Counter check box is selected for the ZRSCSIZE variable in the FILE class template.</p> <p>Any variable containing an integer can be accumulated and stored in parent persistent objects by selecting the Manager 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.</p>

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.

Adding 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** to insert the new attribute above the highlighted attribute, or click **Add After** to insert the new attribute after the highlighted attribute. 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 select the **Type** option from the drop-down list, to indicate the desired attribute type. 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. Refer to *Understanding the Resolution Process* starting on page 199.

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 will appear in the **Attribute Description** column of the **Editing Instance** dialog box when editing instances of this class. 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 may be more meaningful for your purposes. For methods and expressions, try to indicate the purpose of the method or expression in the description.

4. To set the value of an attribute as the display name in the Radia System Explorer, 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 Radia System Explorer displays for entering or editing that variable 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 for a flag (i.e., a yes/no field), a set of check boxes for a flag set (i.e., a field consisting of multiple single-character codes), or a drop-down list for variables from which the user must choose a value from a small list of predetermined values.

To enable this feature, the **Use Edit Masks When Editing Instances** check box must be selected on the **Instance Options** tab of the **Options** dialog box. See Table 2.8 on page 36 for details.

Flag Attribute

A flag attribute is a one-character field whose value can be **Y** or **N**. To direct the Radia System Explorer to offer 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 this example:

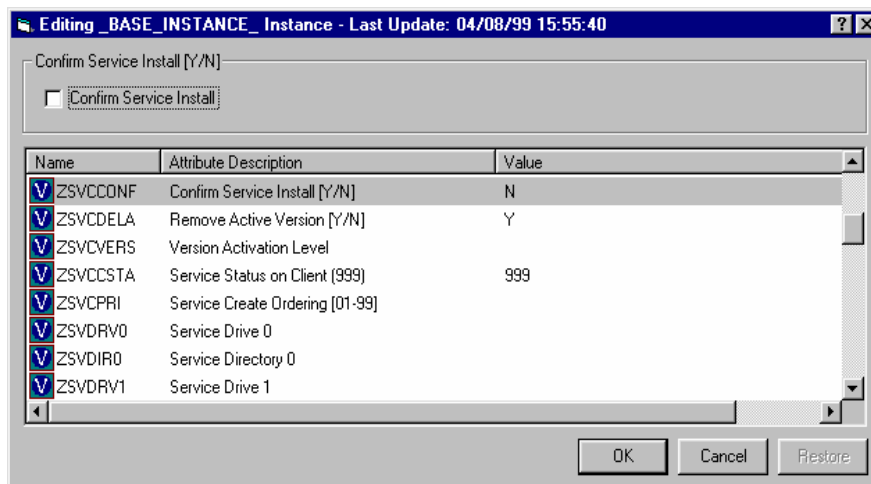


Figure 3.16 ~ The Attribute Description for the ZSVCCONF variable terminates with [Y/N], indicating a flag attribute.

The ZSVCCONF attribute is one character in length, and its Attribute Description, **Confirm Service Install [Y/N]**, terminates with **[Y/N]**. Thus, to enter or edit this variable, the Radia System Explorer offers 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" in the flag set attribute, the character value associated with the flag appears in the flag set string. If a flag is "off" in the flag set attribute, the character value associated with the flag does not appear in the flag set string. An empty flag set string indicates that none of its flags is "on" .

For example, the DOS File Attribute flag set for a particular file is contained in the ZRSCRASH attribute of the FILE class. This four-character attribute holds a string indicating 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 follows:

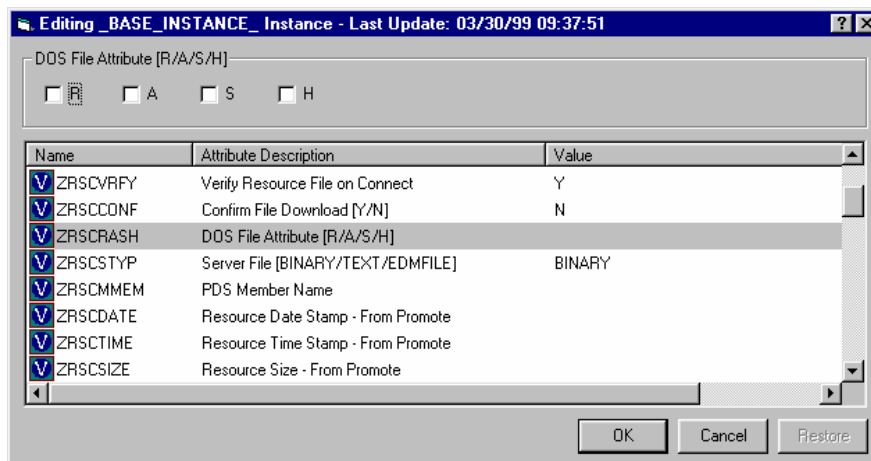


Figure 3.17 ~ An attribute terminating with one-character choices separated by slashes and enclosed in brackets is treated as a flag set.

If the Attribute Description for a multi-character attribute terminates with one-character choices separated by slashes and enclosed within square brackets, and the number of choices equals the length of the attribute (as in the ZRSCRASH attribute, above), the Radia System Explorer treats the attribute as a flag set. It presents 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 Radia System Explorer offers 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, the ZRSCSTYP attribute of the FILE class offers three choices in the Attribute Description, as shown here, after clicking on the drop-down list control:

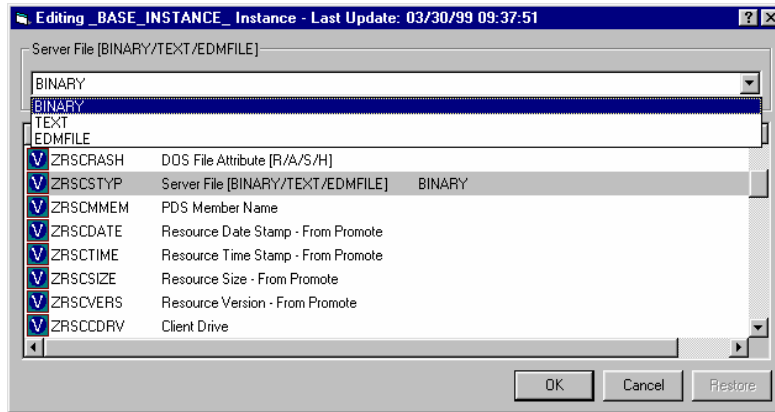


Figure 3.18 ~ An attribute description terminating with a series of choices separated by slashes and enclosed in brackets is offered a drop-down list for editing attributes.

5. As you make changes to a class, you may change your mind and wish to start over. Once a change has been made during class editing, the **Restore** button will be enabled. Click this button to discard changes you've made to the class since the **Editing Class** dialog box opened.

The following message opens.

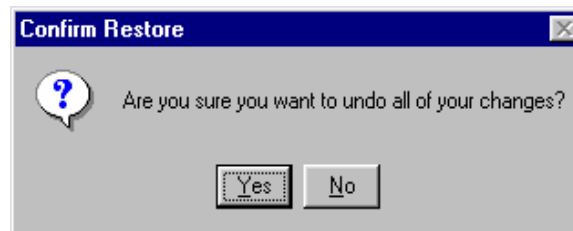


Figure 3.19 ~ Confirm Restore dialog box.

- Click **Yes** to discard changes and begin editing the class from the beginning.
 - Click **No** to dismiss the confirmation without discarding your changes.
6. When you've completed editing the class, click **OK** to save your changes and exit the **Editing Class** dialog box.

7. If there are blank attribute names or descriptions in the class, or if an attribute's length is zero, the following warning will appear.

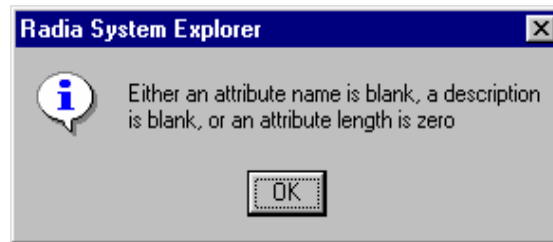


Figure 3.20 ~ Warning dialog box for blank attribute fields.

8. If there are no blank attribute names or descriptions in the class, and no attribute's length is zero, a change confirmation message opens.

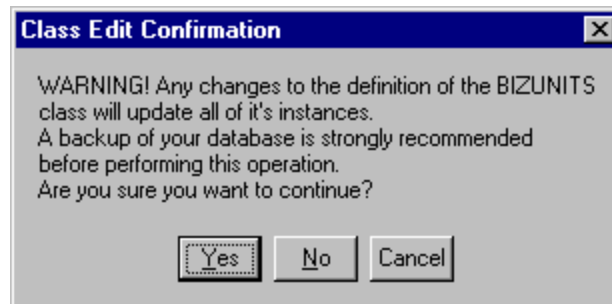


Figure 3.21 ~ Class Edit Confirmation dialog box.

9. 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 Radia System Explorer tree view.
2. Select **Delete Class** from the shortcut menu that opens.

A confirmation dialog box opens.

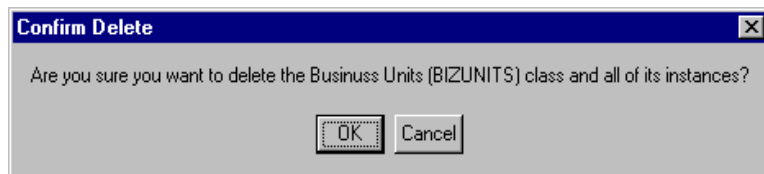


Figure 3.22 ~ Confirm Delete dialog box for deleting a class.

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

Adding an Instance

New instances can be added by copying an existing instance, or adding a new instance from scratch. Which to choose depends upon whether an instance exists whose attribute values are substantially similar to the instance you want to add. If so, copy the exiting 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.

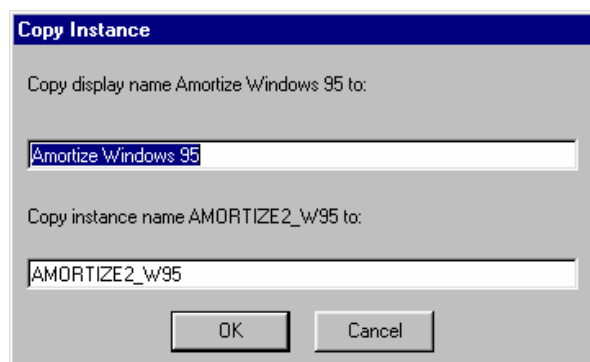


Figure 3.23 ~ Copy Instance dialog box.

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 Radia System Explorer 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 Radia System Explorer window, when the instance is selected and opened in the tree view.

Adding an Instance from Scratch

Adding a new instance from scratch creates a new instance with attribute values set as in the base instance of the class.

To add a new instance from scratch

1. Choose **New Instance** from a shortcut menu.

A **Create Instance** dialog box opens.

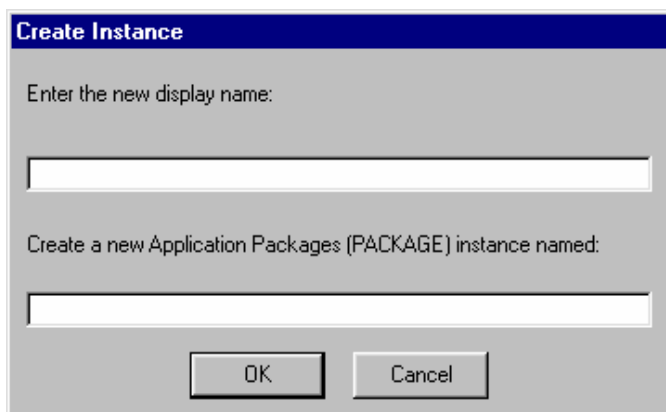


Figure 3.24 ~ Create Instance dialog box.

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 Radia System Explorer 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 Radia System Explorer window.

You may now edit the new instance to contain attribute values intended for the new instance.

Editing an 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 Radia System Explorer window.

The **Editing Instance** dialog box opens.

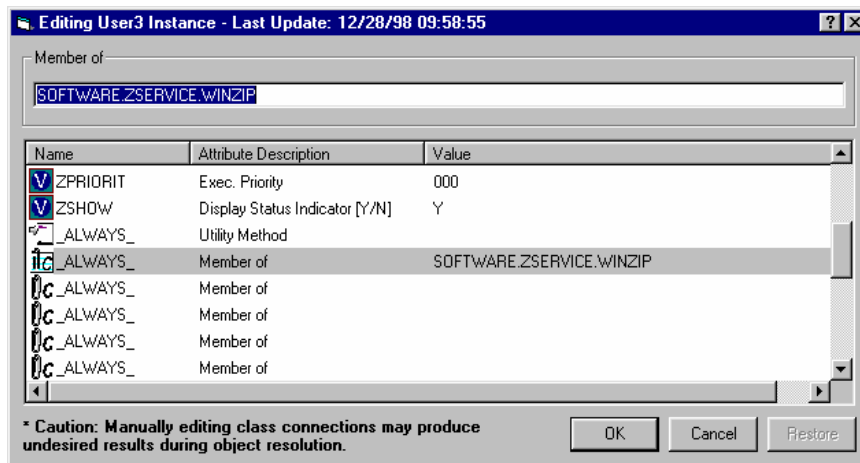


Figure 3.25 ~ Editing Instance dialog box.

If you chose **Edit Instance** from a shortcut menu, the dialog box opens with the first attribute in the instance opened for editing. If you double-click an attribute name to open the dialog box, the attribute whose name you double-clicked is open for editing.

The dialog box lists the attributes comprising 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 67 for information about how the Radia System Explorer determines which edit control to use for a particular attribute, based upon a specification in the attribute's description in the class template.

3. To highlight and select a different attribute, click its name.
4. Continue to select each attribute you want to edit, and to type in the desired value for each, until you have completely edited the instance.

5. Click **OK** to close the dialog box and save your changes.

The following confirmation message opens.

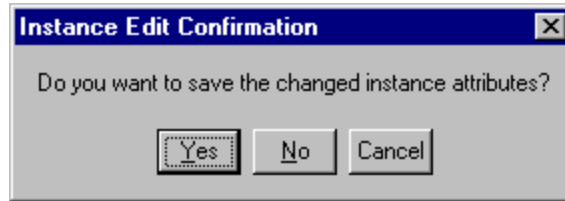


Figure 3.26 ~ Instance Edit Confirmation dialog box.

Note

This confirmation message opens if you have the **Prompt: Before Instances Updated** check box selected in the **Instance Options** tab of the **Options** dialog box. The **Options** dialog box is described in *Setting Preferences: The Options Dialog Box* starting on page 32.

6. Click **Yes** to complete the update of the instance.

Click **No** to discard your changes to the instance, and return to the Radia System Explorer main window.

Click **Cancel** to return to the **Editing Instance** dialog box.

When editing 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. The recommended way to make class connections between attributes is to use the drag-and-drop operation, which is described in *Drag-and-Drop Connections* on page 100.

Editing Multiple Instances

The Radia System Explorer enables you to edit multiple instances in a single operation.

A multiple instance edit operates upon a set of selected instances. 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 of the instances in the selected set. No changes are stored in the Radia Database until you exit the dialog box and confirm the change operation.

To initiate and execute a multiple instance edit

1. Select the instances in the list view of the Radia System Explorer 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
Adaptability Behaviors PGM Discovery	RADIA_DISCOVERY	SOFTWARE.PACKAGE Instance
Office 2000 Standard	O2KSTD	SOFTWARE.PACKAGE Instance
Radia Adaptability 10/18/99	RADIA_ADAPT_101899	SOFTWARE.PACKAGE Instance
Radia Adaptability Behaviors 5/20/99	ADAPT2	SOFTWARE.PACKAGE Instance
Radia Behaviors	RADIA_BEHAVIORS_232	SOFTWARE.PACKAGE Instance
Radia Behaviors 5/18/99	ADAPT	SOFTWARE.PACKAGE Instance
Radia Client Behaviors	CLIENT_BEHAVIORS_237	SOFTWARE.PACKAGE Instance
Sales Demo Windows 95/98	SALES2_W95	SOFTWARE.PACKAGE Instance
Sales Demo Windows NT	SALES2_NT	SOFTWARE.PACKAGE Instance
Sales Update - Microsoft	SALES_MS	SOFTWARE.PACKAGE Instance
Stage All Packages	STAGE	SOFTWARE.PACKAGE Instance
Update and Install Behavior August 10, 1998	CLIENT_BEHAVIOR_081098	SOFTWARE.PACKAGE Instance
Windows Installer	WINDOWS_INSTALLER	SOFTWARE.PACKAGE Instance

Figure 3.27 ~ Initiating a multiple instance edit.

2. Right-click on the selected instance that you want to use as the template for the multiple instance edit operation.

A shortcut menu opens:

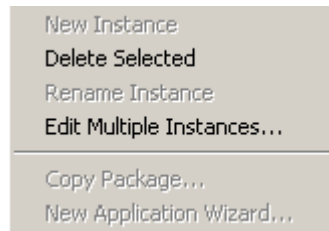


Figure 3.28 ~ Select Edit Multiple Instances.

3. Select **Edit Multiple Instances** from the shortcut menu.

The **Editing Instance** dialog box opens:

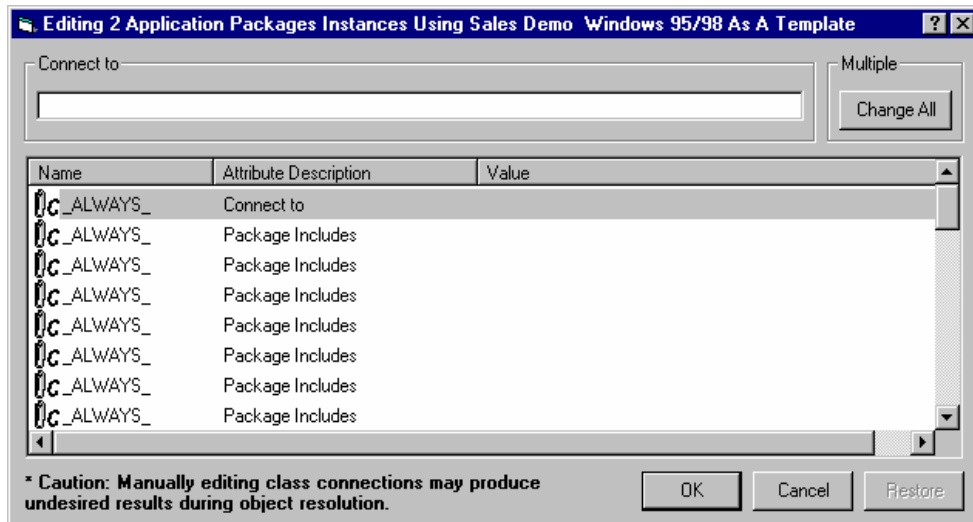


Figure 3.29 ~ Editing Multiple Instances dialog box.

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**:

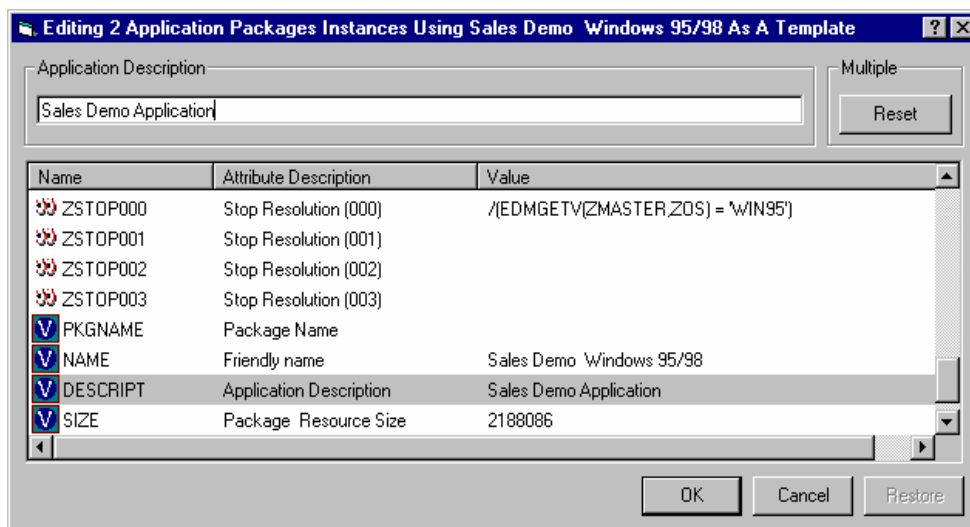


Figure 3.30 ~ The Change All button has been changed to Reset.

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

- Click **Reset**.
As you scroll through the instance's variables, you can determine if you've made a change to the selected variable by referring to the caption on this button.
 - ◆ If the caption is **Change All**, you've made no change to the variable during the current **Editing Instance** dialog box session.
 - ◆ If the caption is **Reset**, you've 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:

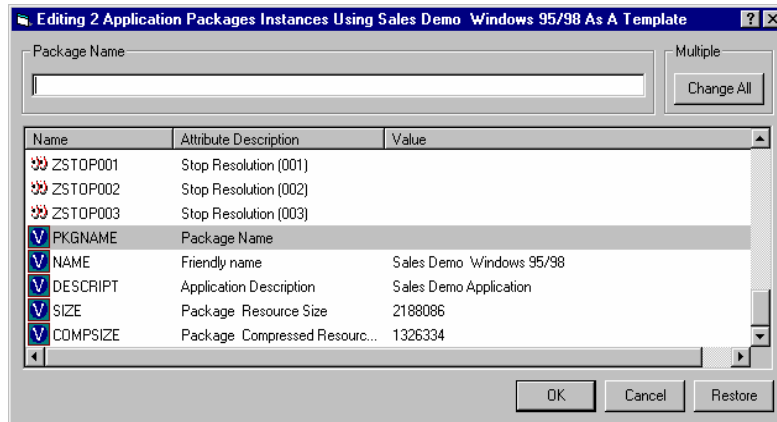


Figure 3.31 ~ The Change All button is active, indicating no change was made to this highlighted variable during the current Editing Instance dialog box session.

To undo all changes you've made in the current Editing Instance dialog box session

1. Click **Restore**.

When you click **Restore**, you are prompted to confirm the operation:

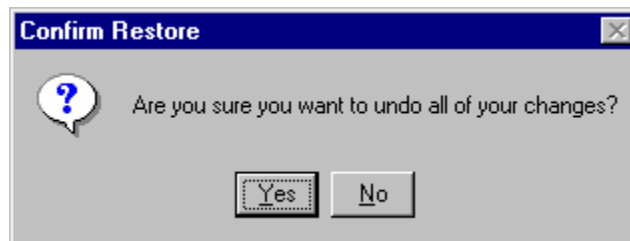


Figure 3.32 ~ The Confirm Restore message.

2. Click **Yes** to discard all changes you've made in the current **Editing Instance** dialog box session.

Click **No** to resume the **Editing Instance** dialog box session without restoring original values.

Changes made during the **Editing Instance** dialog box session are not saved in the Radia Database until you click **OK**. You are prompted to confirm your changes:

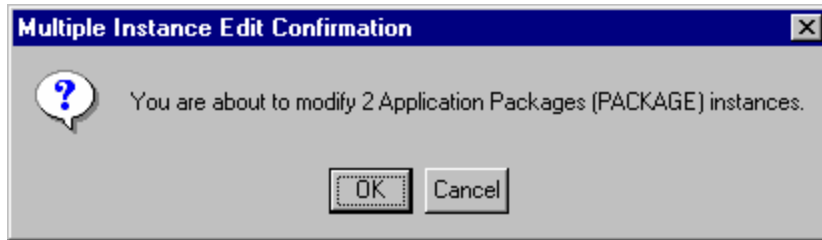


Figure 3.33 ~ Multiple Instance Edit Confirmation dialog box.

3. Click **OK** to save the changes you've made in the template instance, to all of the selected instances, into the Radia Database.

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 to the Radia Database.

Renaming an Instance

Be careful when renaming instances. Instances are connected together 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 to be renamed.

A shortcut menu opens.

2. Select **Rename Instance** from the shortcut menu.

A **Rename Instance** dialog box opens.

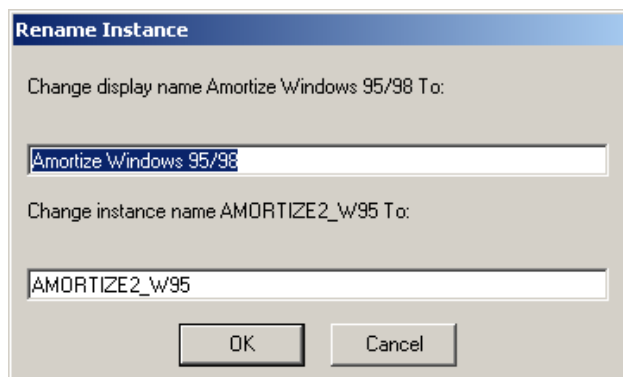


Figure 3.34 ~ Rename Instance dialog box.

3. Enter a new display name (i.e., 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 affects 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 to be deleted.

A shortcut menu opens.

2. Select **Delete Instance**.

A confirmation message opens.

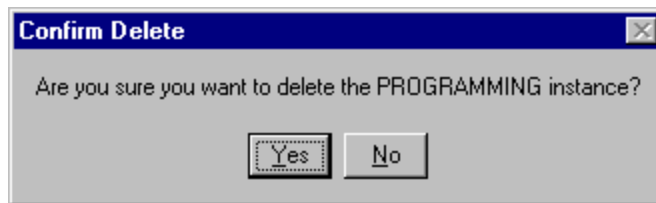


Figure 3.35 ~ Confirm Delete dialog box.

3. Click **Yes** to delete the instance.

Click **No** to close the dialog box without deleting the instance.

Summary

- The Radia Database stores your distribution model.
- The Radia Database is hierarchically organized and subdivided into files, domains, classes, and instances.
- The tree view of the Radia System Explorer tracks the navigation history during an open session.
- Use compound names to help you manage large numbers of instances within the Radia Database.
- The base instance of each class provides default values for instances in that class. Any attribute in an instance will assume the value for that attribute found in the base instance of the class unless a specific value has been entered for that attribute in the instance.
- Use the Radia System Explorer shortcut menus to help you add, edit, copy, rename, and delete Radia Database components.

Building a Distribution Model for Your Radia Database

At the end of this chapter, you will:

- Understand the Policy and Software domains.
- Be able to create instances of the Policy domain.
- Understand how to develop policies that meet your organizational needs.
- Understand the various ways of connecting policy instances to services.
- Be familiar with the Radia Policy Manager and its uses.

Building a Distribution Model

Instances in the Radia Database, the values they contain, and the connections among them define the distribution model for your enterprise. Most of the administration of Radia begins in two domains:

- Policy domain
- Software domain

The Policy Domain

Each client computer managed by Radia is represented by an instance of the POLICY domain Users class. It is used to organize your subscribers into logical groups in preparation for distributing software.

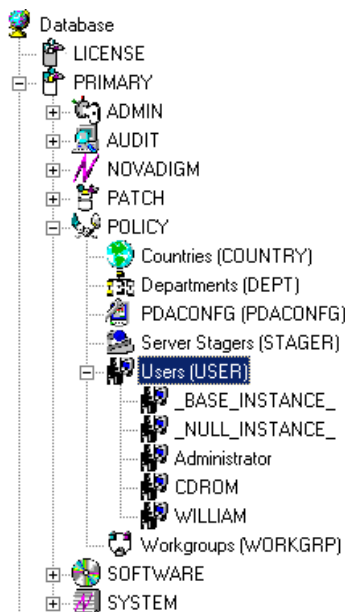


Figure 4.1 ~ PRIMARY.POLICY.Users class.

The POLICY domain has six default classes, Countries (COUNTRY), Departments (DEPT), PDACONFIG (PDACONFIG), Server Stagers (STAGER), Users (USER), and Workgroups (WORKGRP), as described below.

Table 4.1 ~ Policy Domain

Class	Description
Countries (COUNTRY)	Use for clock synchronizations with the Radia Configuration Server. Do not assign services to this class.
Departments (DEPT)	Use to group subscribers into departments.
PDACONFG (PDACONFG)	Used to define the parameters for PDA configuration. The PDACONFG class gives the administrator a way of targeting and storing PDA-specific information during resolution.
Server Stagers (STAGER)	Use to define Radia Staging Servers within your distribution network. Also, use to define storage location on a Radia Staging Server computer.
Users (USER)	Use to define individual subscribers. If you would like to create policies at the user level, then you must have an instance in the USER class for each subscriber.
Workgroups (WORKGRP)	Use to group subscribers into functional groups. For example, a project team may be made up of subscribers from several different departments.

Users can be grouped by connecting them to instances of other POLICY domain classes such as Departments or Workgroups classes. This allows Radia to manage content on client computers based upon the user belonging to a workgroup, or being a member of a department.

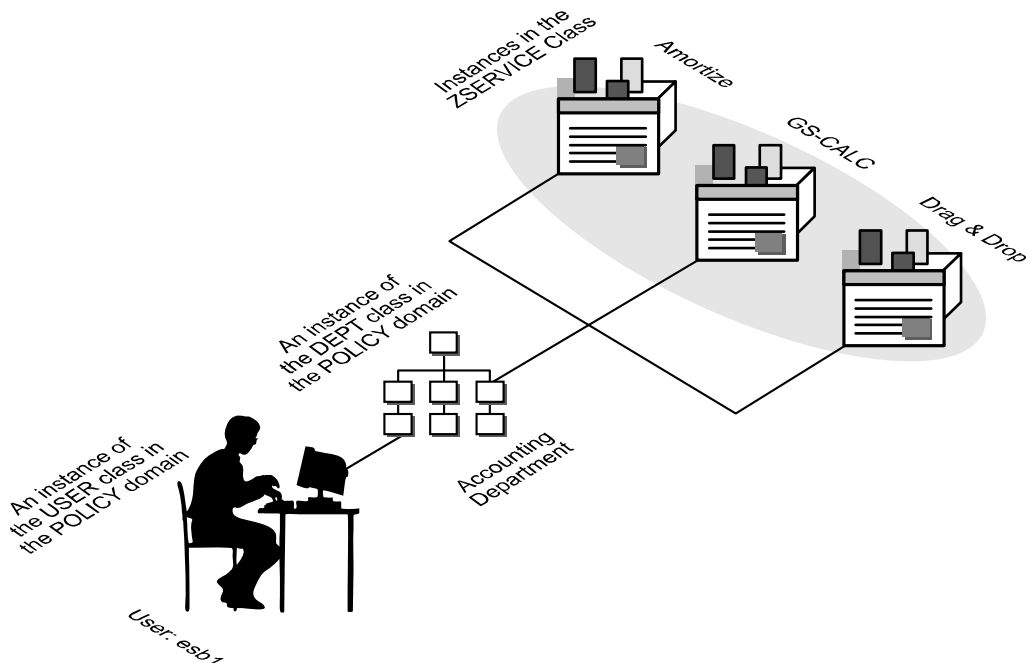


Figure 4.2 ~ Create a Policy for your organization.

You'll probably want to begin developing your distribution model by creating new users.

To create a new user

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box opens.

Note

The **User ID**, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**.
3. The Radia System Explorer splash screen appears briefly.
4. Double-click **PRIMARY**.

5. Double-click **POLICY**.
6. Double-click **Users (USER)** to expand the class. A list of all of the User instances opens.
7. Right-click **Users**. A shortcut menu opens.

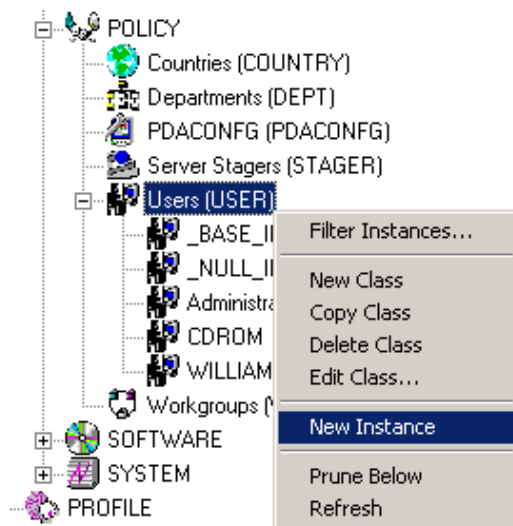


Figure 4.3 ~ To create a new user, right-click the Users class. Select New Instance from the shortcut menu.

8. Select **New Instance** from the shortcut menu. The **Create Instance** dialog box opens.

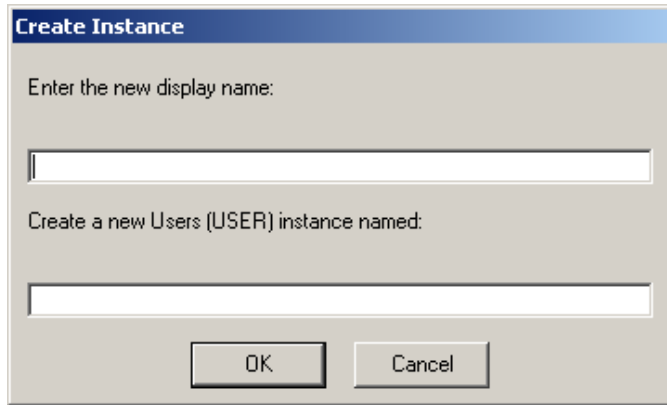


Figure 4.4 ~ Create Instance dialog box.

9. Enter the display name of the new User. This is the friendly name as it will appear in the tree view. As an example, create a new user instance, ESB1.
 10. Enter a user's instance name.
-

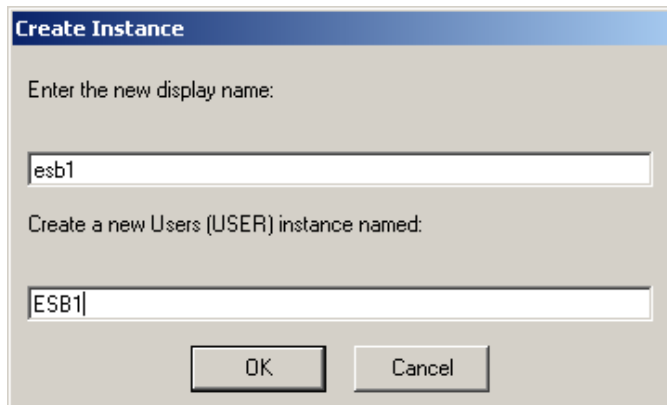


Figure 4.5 ~ Completed Create Instance dialog box.

11. Click **OK**.

A new User instance, esb1 is created.

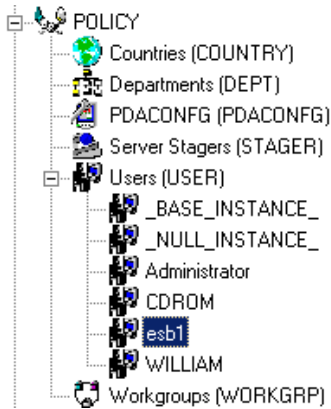


Figure 4.6 ~ A new user, *esb1*, has been added to the Users class.

Use the Radia System Explorer to create, copy, delete, rename and edit user instances that suit your enterprise needs.

Software Domain

An instance of the SOFTWARE domain Application Packages (PACKAGE) represents one software application or unit of other content to be managed on client computers. The Radia Database is shipped with sample packages as shown in Figure 4.7 on page 92.

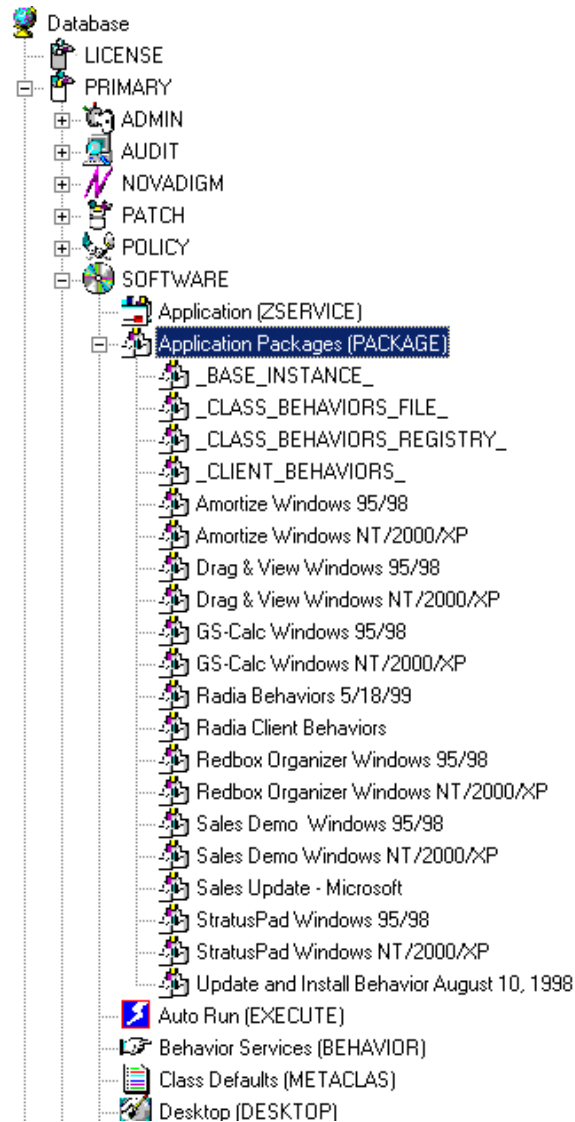


Figure 4.7 ~ Application Packages (PACKAGE) class

Using the Radia Packager, the administrator packages and publishes content (software, text files, and so forth) to the Radia Database. These packages are stored in the Application Packages (PACKAGE) class.

Once the package has been created, the administrator will need to create an Application (ZSERVICE) for the package to be deployed to end users. This is referred to as creating a Service. The Radia Database ships with sample Services for the packages within the Package class.

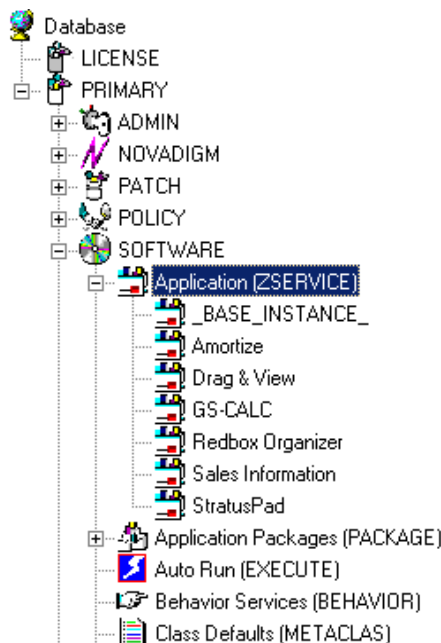


Figure 4.8 ~ Sample Application (ZSERVICE) instances as shipped from HP.

For information on packaging and publishing application packages, see the *Radia Application Manager Guide* or the *Radia Software Manager Guide*. For information on creating a ZSERVICE, see *Running the New Application Wizard* beginning on page 120.

The ZSERVICE instance that represents a unit of managed content can be connected to many other class instances. Those class instances can then be connected further to other class instances. If, for example, the unit of managed content is a software application, each file in the application will be represented by an instance of the FILE class. In addition, there can be instances of the REGISTRY class that represent an application's registry keys, instances of the DESKTOP class representing the links and shortcuts that invoke the application once it is installed, and so forth. The distribution model for a complex application such as Microsoft Word can consist of several thousand connected instances within the Radia Database.

The Radia System Explorer graphically shows the connections among instances in the tree view. This enables the Radia administrator to work with instances that are connected to one another in a localized portion of the tree view.

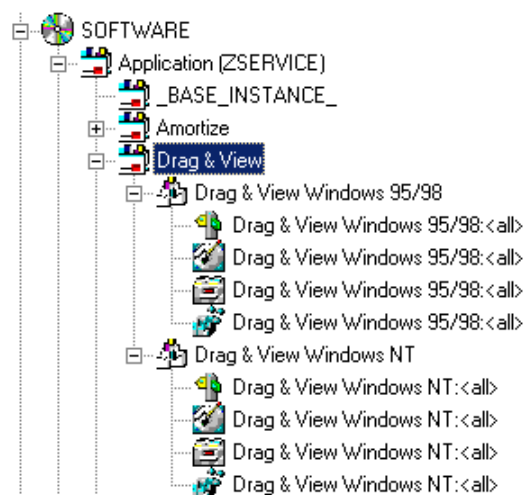


Figure 4.9 ~ An example of the Drag & View service and its connections.

Inter-Package Relationships

Relationships between packages are established by the kind of connection that exists between the PACKAGE class instances in the Radia Database.


















 C_ALWAYS_	Package Includes
 C_ALWAYS_	Package Includes
 C_ALWAYS_	Package Includes
 C_ALWAYS_	Package Includes
 C_ALWAYS_	Package Includes
 I INCLUDES	Package Includes
 I INCLUDES	Package Includes
 I INCLUDES	Package Includes
 I INCLUDES	Package Includes
 I INCLUDES	Package Includes
 I INCLUDES	Package Includes
 R REQUIRES	Required Package Level
 R REQUIRES	Required Package Level
 R REQUIRES	Required Package Level
 R REQUIRES	Required Package Level
 R REQUIRES	Required Package Level
 R REQUIRES	Required Package Level

Figure 4.10 ~ Each variable can hold a connection to another package.

Each variable in the above example can hold a connection to another package.

Two packages can be related such that either one package *Requires* the other, or one package *Includes* the other.

Requires Connection

When one package requires another, the components of the requiring package supercede the same components in the required package. For example, if package A requires package B, the components in A supercede any of the same components in B. This relationship is useful for deploying maintenance to installed application software, where Maintenance Package A requires the Base Package B.

Consider the following example of applying the Service Release 1 update to the Microsoft Office application suite.

When Office was introduced, there was no Service Release 1 update. The Radia administrator packaged the components of the Office suite for Windows 95 and Windows NT. Later, when Microsoft issued Service Release 1 update for Office, the Radia administrator was able to create new packages for each component and for each operating system which representing the difference between the base install of the component and the Service Release 1 update. This was done by having the Service Release 1 Update require the packaged base install of the Office suite for Windows 95 and NT.

Includes Connection

The ability to create a relationship between Radia packages such that one package includes another package enables the Radia administrator to deploy application software constructed from multiple Radia packages.

When one package includes another, both are deployed.

A typical use for this capability arises in installations where the application software to be deployed must be accompanied by customized extensions.

For example, your organization may have a series of customized Microsoft Word templates that need to be deployed along with Word. The Radia administrator can package Microsoft Word in a separate package for each target platform, and package the customized templates in a separate package of their own. Using the Radia System Explorer, the administrator would create an Includes connection in each of the Microsoft Word package instances, connecting to the package instance representing the customized templates. Microsoft Word would then be deployed with the customized templates whenever a subscriber installs Microsoft Word.

Determining Policies

After using the Radia Packager and Radia System Explorer to create services, you must determine who gets what. In other words, you must set up policies that identify which user receives what software.

Before you set up policies in Radia, you will want to have a carefully constructed plan in place. Consider the following:

- What group policies do you need to create?
- What groups should your users be assigned to?
- What software will Radia manage for the users?

Later, you will assign each USER instance to one or more groups, as necessary. Policies are easier to administer when subscribers are grouped according to the content that Radia will manage for them.

For example, your sales department may have 50 employees. Rather than authorizing each user for a spreadsheet application, you might want to create an instance in the Departments class called Sales. Then, you can group the 50 employees into the Sales department instance, allowing you to authorize applications, or even take away authorization to applications, for all 50 employees simultaneously.

In the next example, you will create a Departments (DEPT) instance called Sales.

To create a Departments (DEPT) instance

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box appears.

Note

The **User ID**, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**.
3. The Radia System Explorer splash screen appears momentarily.
4. Double-click **PRIMARY**.
5. Double-click **POLICY**.

6. Double-click **Departments (DEPT)** to expand the class.
7. Right click on the **Departments (DEPT)** class. A shortcut menu opens.

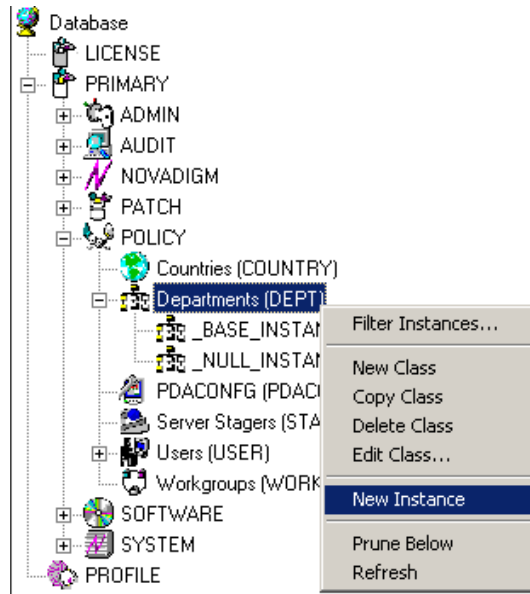


Figure 4.11 ~ Select New Instance from the shortcut menu.

8. Select **New Instance**.
The **Create Instance** dialog box opens.
9. Enter the display and instance names in the appropriate fields. For this example, Sales.

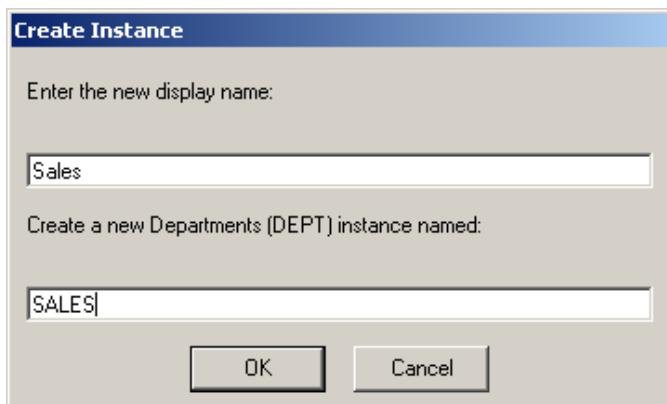


Figure 4.12 ~ Create Instance dialog box.

- 10.** Click **OK**. The Sales Department is added.
-

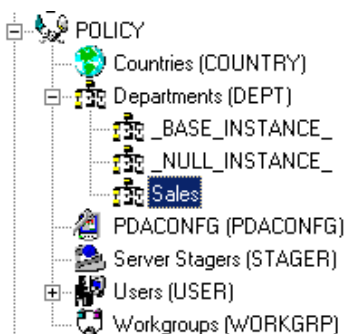


Figure 4.13 ~ The newly created Sales Department in the Departments (DEPT) class.

The next example shows how to connect your ESB1 user to the Sales Department instance.

Connection Types

The actual practice of connecting Radia Database instances to one another can be accomplished in several ways.

Manual Connections

You may choose to set the connections between instances manually by typing the path of the connection into a **Connection** attribute. See Table 3.2 on page 63 for additional information regarding connection types.

Caution

While this method is supported, it is recommended that your connections be completed using drag-and-drop or the **Show Connections** dialog box as indicated below.

Drag-and-Drop Connections

The Radia System Explorer features the ability to make drag-and-drop connections between instances of connectable classes. Using this feature will avoid two common errors that occur when you type in connection specifications between instances:

- Typographical error
- Typing in a connection to an instance of a class with which a connection is prohibited.

You can also use drag-and-drop connections between multiple Radia System Explorer windows.

The drag-and-drop method is useful when the instances to be connected are close enough to be displayed in the tree view at the same time.

Invoking a drag-and-drop method can be performed rather easily by selecting one instance (clicking and holding down the left mouse button), dragging it to your target instance, and dropping it (releasing the left mouse button) onto your target instance. After you begin the dragging process, you will note that the cursor changes as you move up and down the tree view.

The circle with a slash indicates that a connection to that particular instance is not possible.

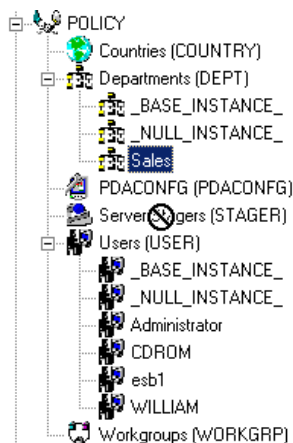


Figure 4.14 ~ The circle with a slash indicates that a connection is not allowed.

A paperclip indicates that a connection is allowed to that instance.

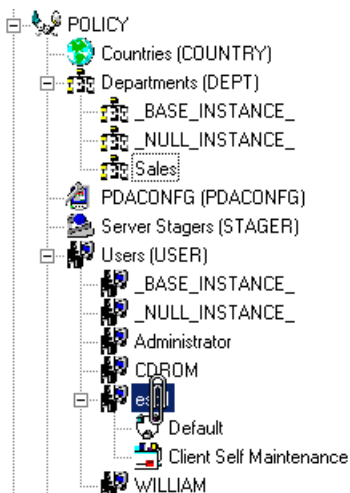


Figure 4.15 ~ The paperclip indicates that a connection is allowed.

Once you release the paperclip, the **Select Connection Attribute** dialog box opens.

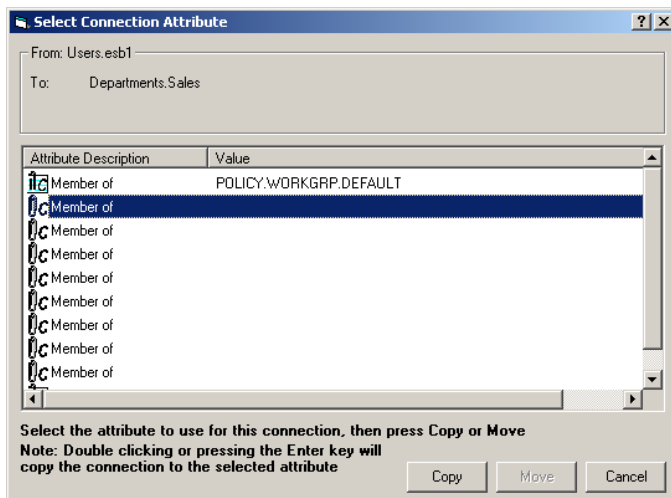


Figure 4.16 ~ The Select Connection Attribute dialog box showing the next possible connection.

The Radia System Explorer highlights the first eligible connection attribute in the selected instance, as shown in Figure 4.16 above.

Press Enter, or click **Copy** to accept this connection attribute. To use a different available connection attribute, scroll to it and double-click it.

In the Radia System Explorer's tree view, an instance containing connections lists those connected instances in the tree immediately beneath itself. When you make a drag-and-drop connection by dragging such a connected instance, you have the option to move the connection from the instance from which the connection was dragged.

Click **Move** to move the connection. The connection specification will be removed from the instance from which it was dragged, and will be stored in the instance on which it was dropped.

If, instead, you click **Copy**, the connection specification will be left in the instance from which it was dragged, and a copy will be stored in the instance on which it was dropped.

After confirming the connection, the connection is complete.

Show Connections

Use the **Show Connections** dialog box when the instances you wish to connect are not close enough to be displayed in the tree view at the same time.

This method of connection is available by right-clicking an instance and choosing **Show Connections** from the resulting shortcut menu.

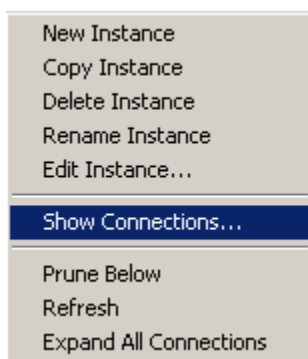


Figure 4.17 ~ Select Show Connections to open the Show connectable classes dialog box.

When using this feature, the **Show connectable classes** dialog box opens.

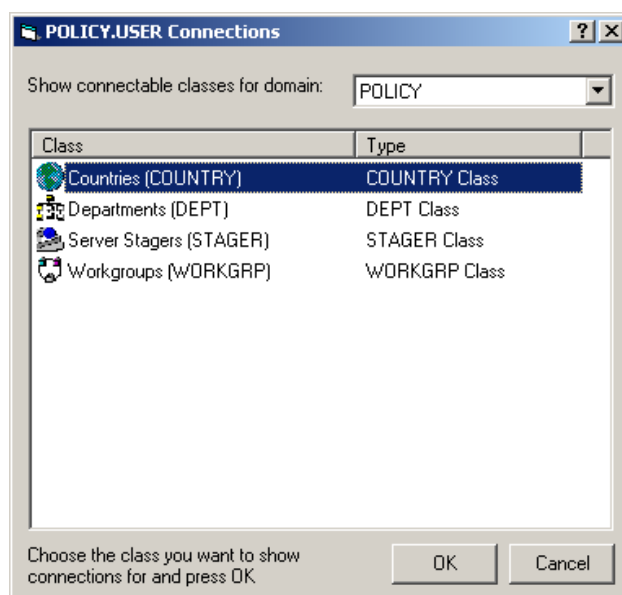


Figure 4.18 ~ Show connectable classes dialog box.

The dialog box lists the classes in the same domain to which connection of the selected instance is permitted.

To connect to a class in a different domain, click the drop-down list labeled **Show connectable classes for domain** and a list of domains appears.

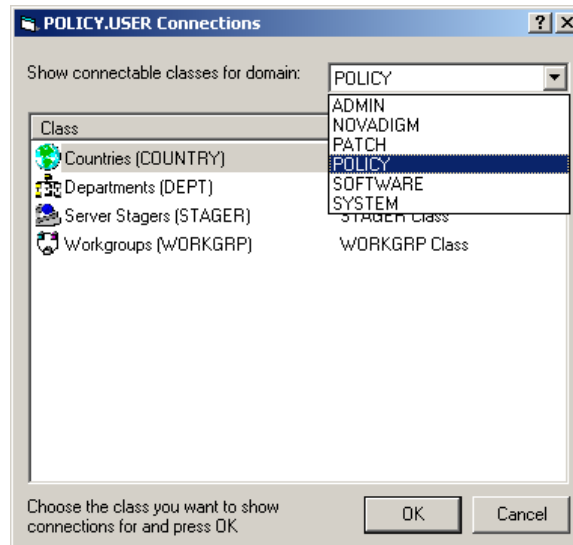


Figure 4.19 ~ Domain list for connectable classes.

If there are no permitted connectable classes in a selected domain, a warning message appears.

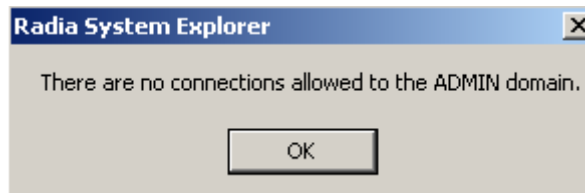


Figure 4.20 ~ Warning message for a non-permitted connection.

Rules for which classes can be connected are contained in the Radia Database. See *Maintaining Connection Rules in the Radia Database* beginning on page 211 for details on how to maintain these rules.

Once an acceptable class has been selected, all instances for that class are displayed in the list view.

Now you can select the class instance from the list view (on the right), and drag-and-drop it onto the target connection instance (on the left).

To connect a User to a Departments (DEPT) instance

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box appears.

Note

The **User ID**, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**.
3. The Radia System Explorer splash screen appears momentarily.
4. Double-click **PRIMARY**.
5. Expand the **POLICY** domain.
6. Double-click **Users (USER)** to expand the class.

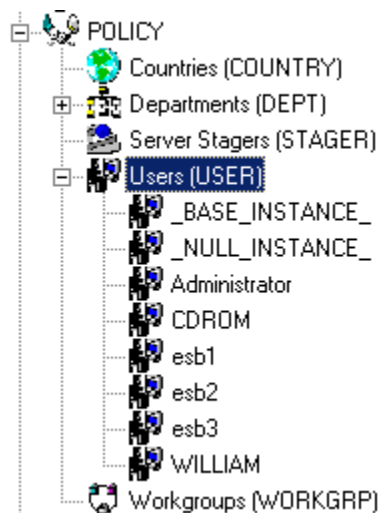


Figure 4.21 ~ Expanded PRIMARY.POLICY.Users (USER) class.

7. Right-click the **esb1** user instance. A shortcut menu opens.

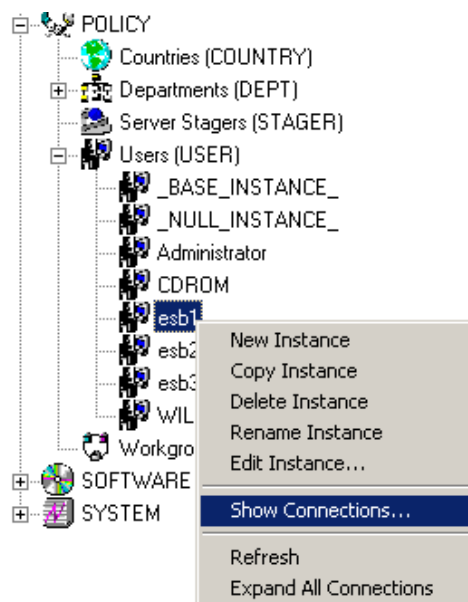


Figure 4.22 ~ Select Show Connections from the shortcut menu.

8. Select Show Connections.

The **Show connectable classes** dialog box opens.

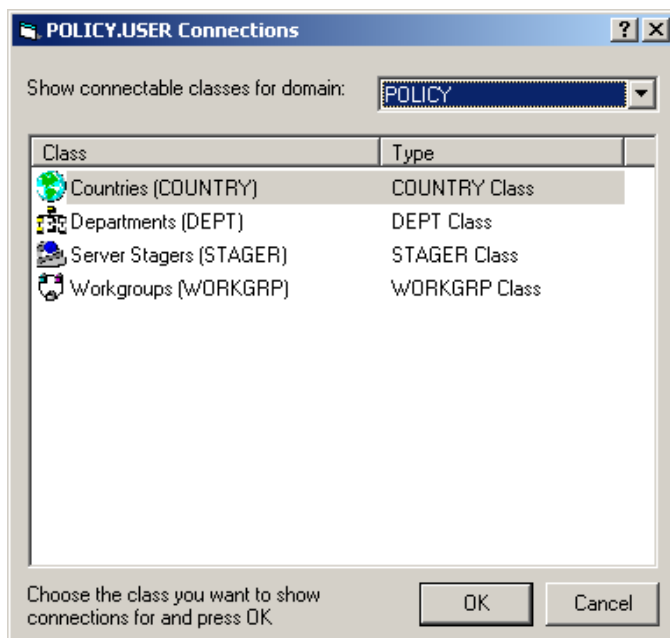


Figure 4.23 ~ Show connectable classes for domain dialog box.

9. Select **Departments (DEPT)** class and click **OK**.

All Departments (DEPT) class instances are displayed in the list view.

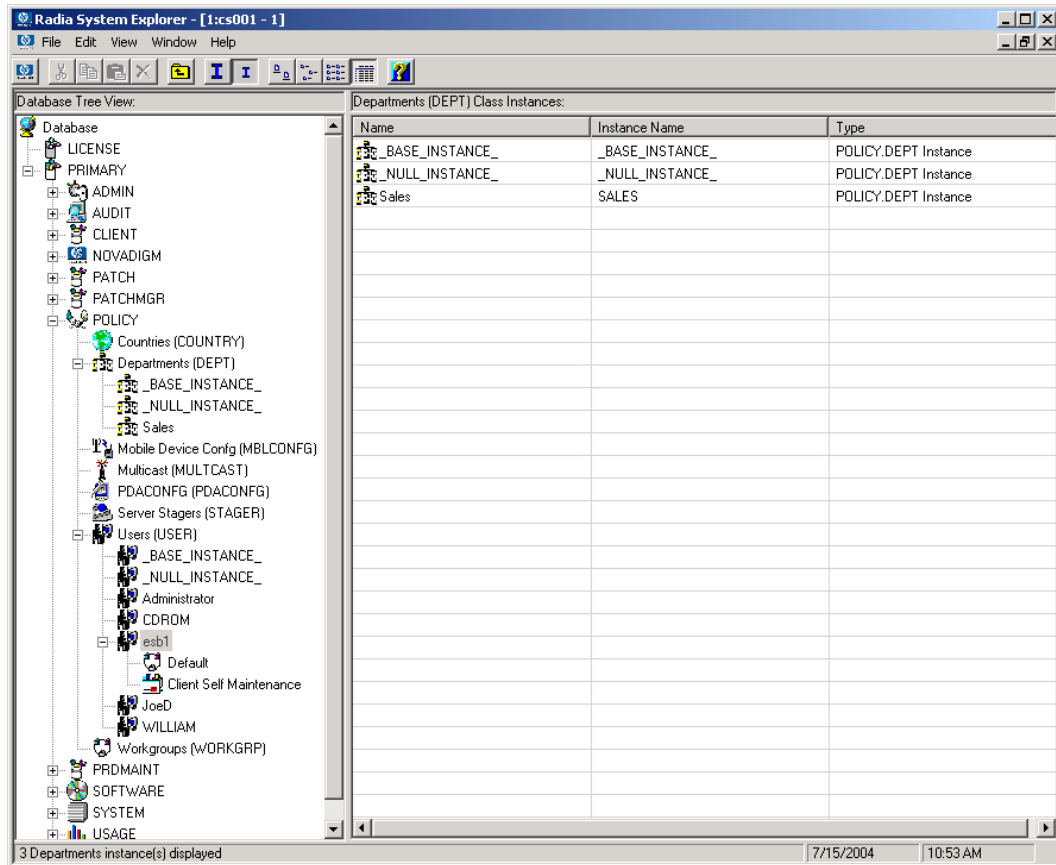


Figure 4.24 ~ All of the Departments (DEPT) class instances are now displayed.

10. Select the Sales instance from the list view, and drag-and-drop it onto the USER esb1 in the tree view. The **Select Connection Attribute** dialog box opens.
 11. Click **Copy** to complete the connection. The **Confirm Connection** dialog box opens.
 12. Click **No** to cancel the connection process.
- Click **Yes** to confirm the connection. An **Instance Connection** dialog box opens.

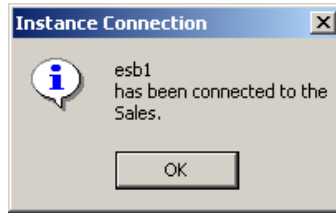


Figure 4.25 ~ Instance Connection confirmation box.

Drag-and-Drop Component Class Instances

No matter what method you use, an additional option is available in the **Select Connection Attributes** window when you drag an instance of a Component class onto an instance of the PACKAGE class. Examples of Component classes include FILE, DESKTOP, and REGISTRY.

Following is an example of the **Select Connection Attributes** window that results from dragging a FILE instance onto a PACKAGE instance.

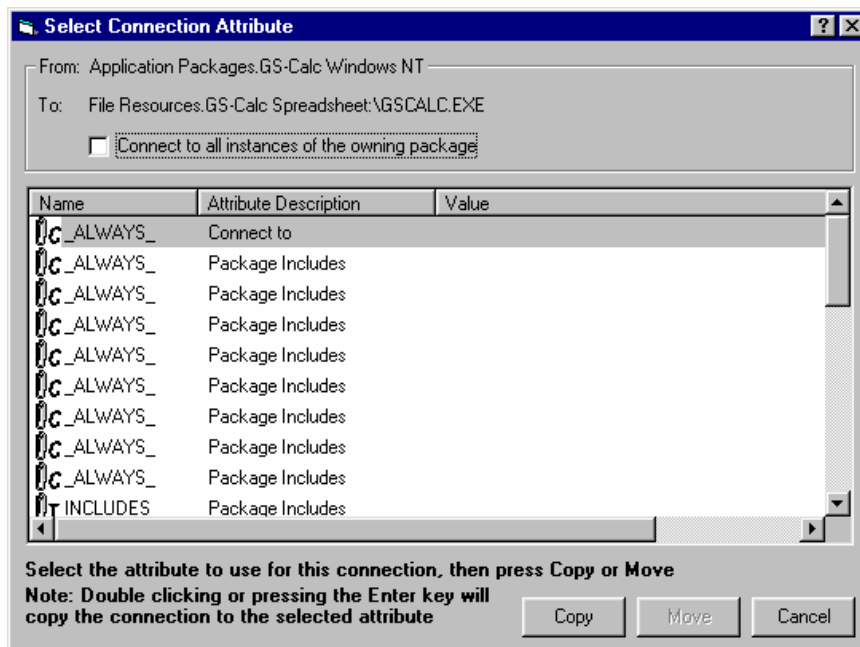


Figure 4.26 ~ Select Connection Attribute dialog box that opens after dragging a FILE instance onto a PACKAGE instance.

An additional option is the **Connect to all instances of the owning package** check box. When you select this check box, the connection specification stored in the PACKAGE instance will be wildcarded to include all of the dragged Component class instances (in this example, FILE instances) that were published together (by Radia Packager) in the same application as the dragged instance. In this example, all of the GS-CALC FILE instances would be connected in a single connection specification, if the check box were selected.

Leaving this check box cleared results in a connection specification that refers only to the instance that was dragged. In the example, this would be the GS-Calc Spreadsheet: \GSCALC.EXE instance.

Connecting Services to Policies

Once you have determined your policies and set up your users and groups, you can begin to entitle each user or group to the applications to be deployed to them. To do so, you will connect **POLICY** class instances to instances in the Application (**ZSERVICE**) class.

In the following example, you will connect the Sales Department Policy to the Amortize Application (**ZSERVICE**).

To connect a service to a policy

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box appears.

Note

The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**.
3. The Radia System Explorer splash screen appears momentarily.
4. Double-click **PRIMARY**.
5. Expand the **POLICY** domain.
6. Double-click **Departments (DEPT)** to expand the class.
7. Right-click the **Sales** instance (in the tree view) and select **Show Connections**. The **POLICY.DEPT Connections** dialog box opens. This dialog box displays a list of classes that you can connect the selected instance to.
8. From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.

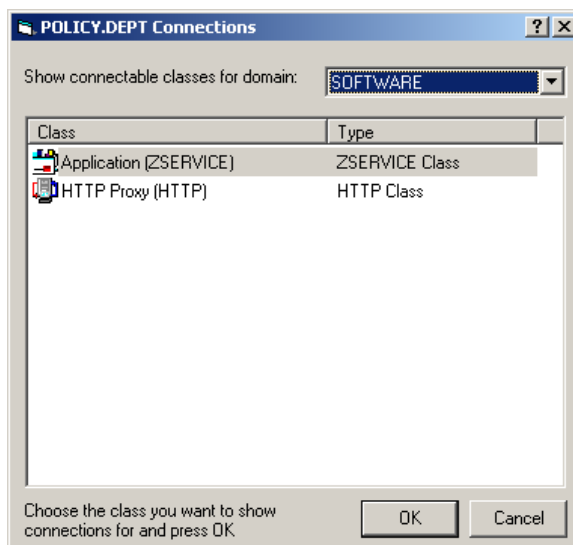


Figure 4.27 ~ Select SOFTWARE from the drop-down list.

9. Click **Application (ZSERVICE)**, and then click **OK**. The instances in the ZSERVICE class appear in the list view.

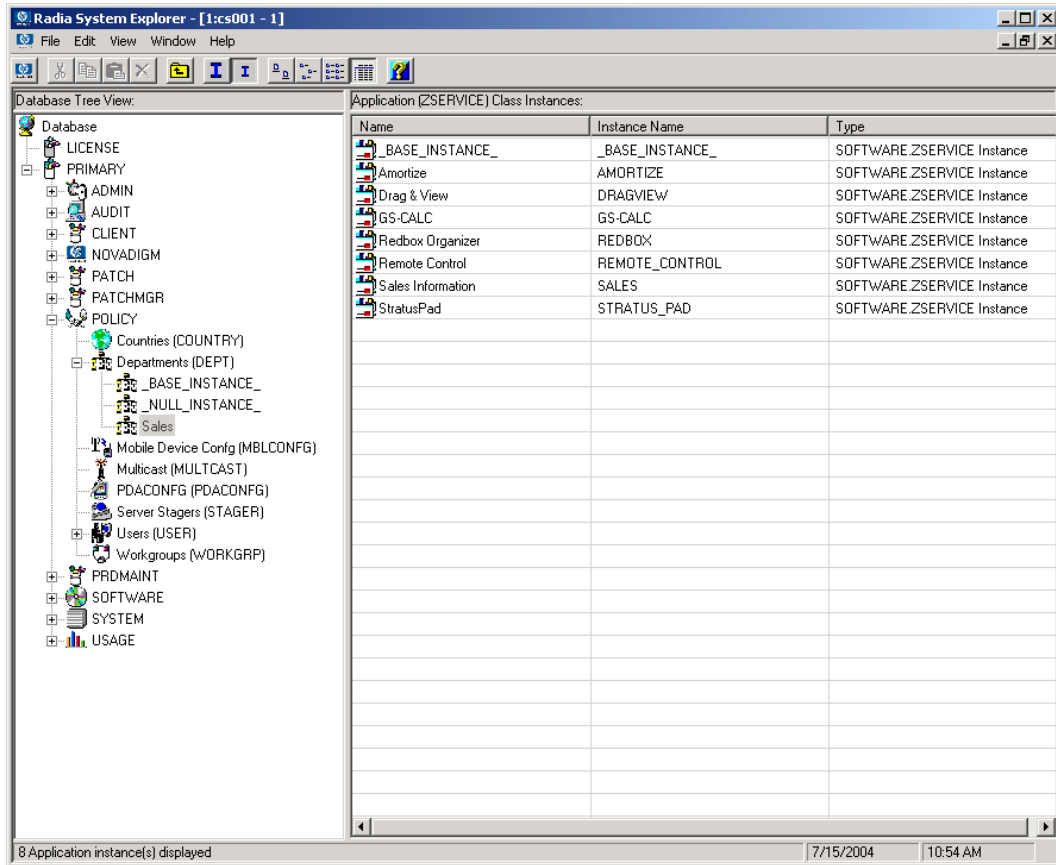


Figure 4.28 ~ The ZSERVICE in the list view.

10. Select the **Amortize** instance from the list view and drag it to the appropriate Departments instance (in this example, **Sales**). When your cursor turns into a paper clip, release the mouse button.

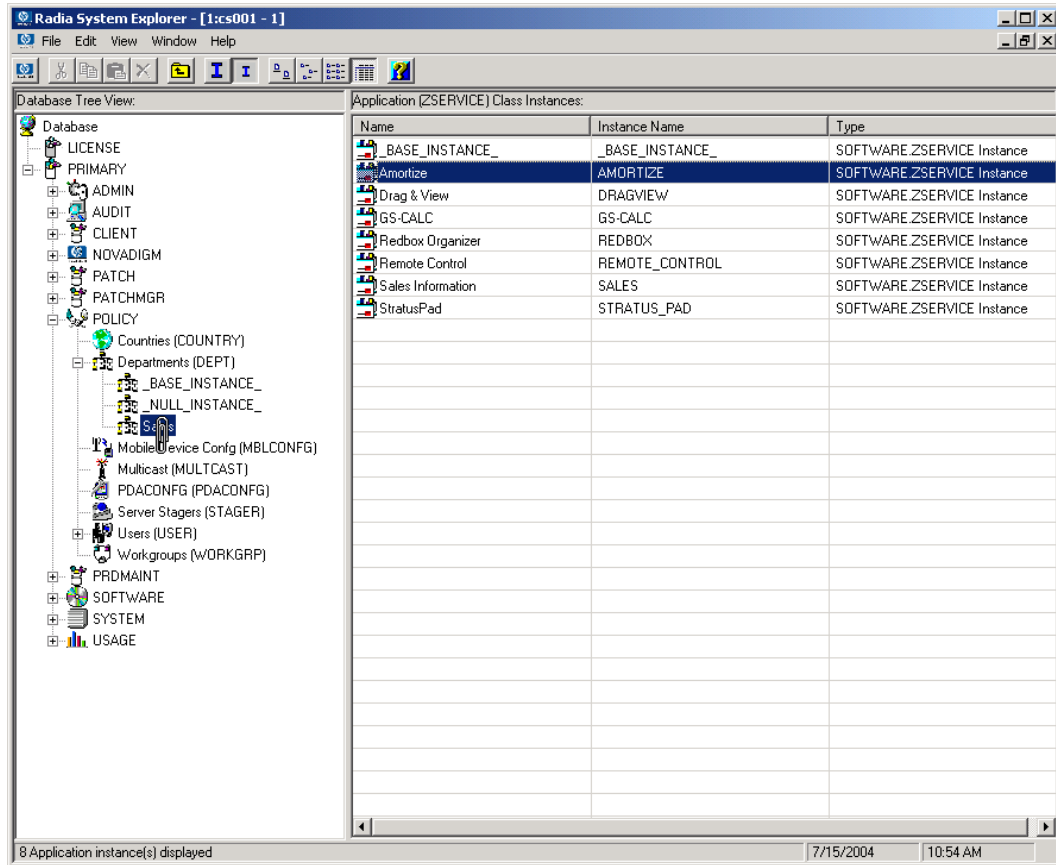


Figure 4.29 ~ Select Amortize and drag-and-drop it onto the Sales Departments (DEPT) instance.

11. The **Select Connection Attribute** dialog box opens.
12. Click **Copy** to create the connection from the Sales department to Amortize.
13. Click **Yes** to confirm the connection.
14. Click **OK** when you receive the confirmation that " Sales has been connected to the Amortize."

Notice that Amortize is listed under the Sales department instance, which indicates that the entire Sales department is now authorized to receive the Amortize application.

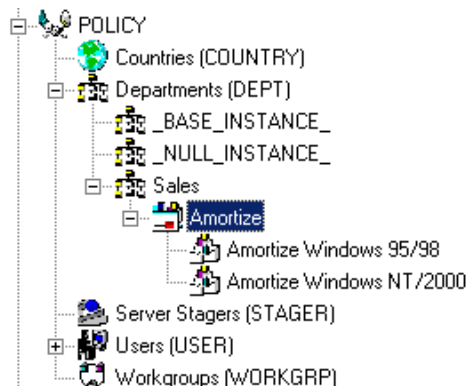


Figure 4.30 ~ All users within the Sales Department (DEPT) will receive the Amortize Service.

A Word about the Radia Policy Server

The *Radia Policy Server* is a Web server used for administrative purposes such as mapping services to users in the directory tree. The *Policy Server*, an add-on to the Radia Configuration Server, determines what services should be distributed and managed for the client that is currently logged on by querying the Policy Server.

The Radia Policy Server plug-in for the Radia Integration Server leverages your investment in directory services while using Radia for software management. This greatly reduces the total cost of ownership of your environment. In other words, directory services handle policy management and Radia manages services. This saves you time because you do not have to define or maintain lists of users in the Radia Configuration Server.

Note

The Radia Policy Server and the Radia Policy Adapter are optional features available with the Radia Infrastructure product set. Contact your HP sales representative for details.

Summary

- Instances in the Radia Database, and the connections among them, define the distribution model for your enterprise.
- Each client computer managed by Radia is represented by an instance of the POLICY domain, Users class.
- Instances of the SOFTWARE domain, Application Packages (PACKAGE) represents one software application or unit of content to be managed on a client computer.
- Instances of the SOFTWARE domain, Application (ZSERVICE) class, represent the PACKAGE resources that will be deployed. A ZSERIVCE instance can contain many Application Packages.
- Define the policies that identify which users receive what software.
- The actual practice of connecting Radia Database instances to each other can be accomplished in several ways: using manual connections, using drag-and-drop connections, and using the **Show Connections** dialog box.
- The Radia Policy Server is a Web server used for administrative purposes such as mapping services to users in the directory tree.

Additional Features of the Radia System Explorer

At the end of this chapter, you will:

- Be familiar with the features of the Radia System Explorer.
- Know how to create services for packages using the New Application Wizard.
- Know how to configure multiple versions of a service using the Version Group Editor.
- Know how to optimize services by creating *patches*.
- Know how to create a Service Group to manage products that require more than one service package to establish full operation.
- Know how to use the Notify capabilities of Radia.
- Know how to filter a large database.
- Know how to build State Files for any application service and have it imported into the Radia Application Knowledge Base.

This chapter describes additional features of the Radia System Explorer, including:

- The New Application Wizard
- The Version Group Editor
- Service Optimization
- Radia Service Groups
- Radia Notify
- Filtering the Radia Database
- Building State Files for export to the Radia Application Knowledge Base

For information on publishing content to the Radia Database, see the *Publishing Applications and Content* chapter in the *Radia Application Manager Guide*.

Running the New Application Wizard

The New Application Wizard automates creation of an Application (ZSERVICE class) instance from a PACKAGE class instance, normally created by the Radia Packager or the Radia Packager for Windows Installer.

The GS-CALC package will be used as an example.

To use the New Application Wizard to create a GS-CALC service

1. Start the Radia System Explorer and navigate to the PACKAGE class of the SOFTWARE domain.
2. Locate the two GS-Calc packages in the tree view (see Figure 5.1 on page 121).
 - One package for Windows 95/98
 - One package for Windows NT

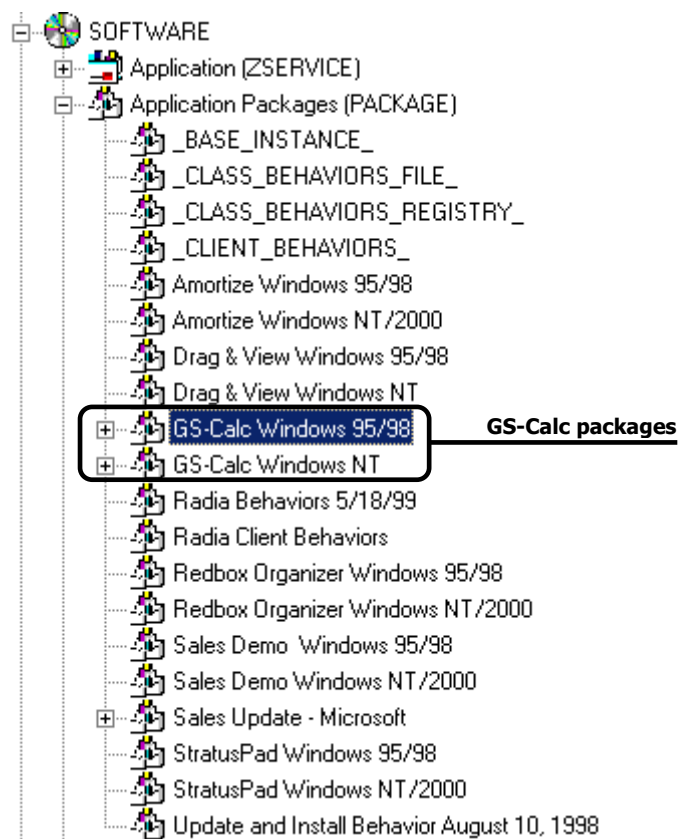


Figure 5.1 ~ The GS-CALC PACKAGE class instance.

3. Right-click on the **GS-Calc Windows 95/98** package class instance (highlighted in the above example).

A shortcut menu opens (see Figure 5.2 on page 122).

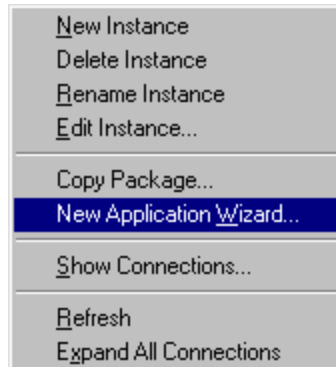


Figure 5.2 ~ Select New Application Wizard from the shortcut menu

- 4. Click New Application Wizard.**

The **New Application Wizard** dialog box opens.

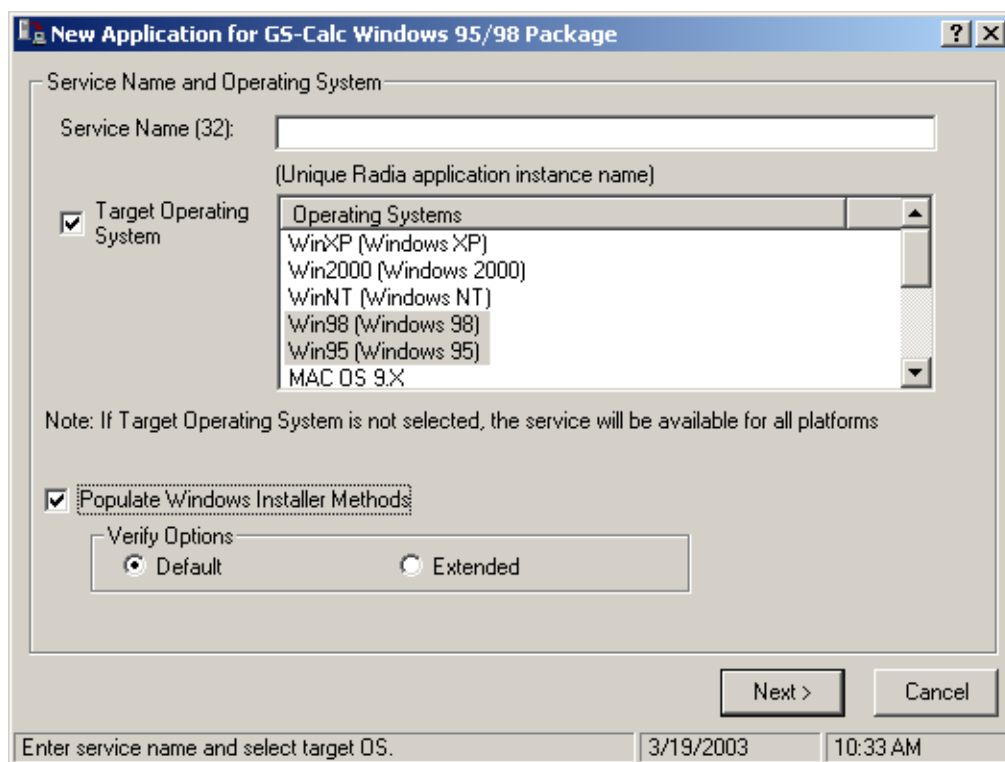


Figure 5.3 ~ New Application Wizard dialog box.

Note

The New Application Wizard dialog box varies for applications created with the Radia Packager for Windows Installer. See the topic *Defining MSI Basic Behaviors on the New Application Wizard* on page 132.

5. Type a name (up to 32 characters) in the **Service Name** text box for the service instance (in the SOFTWARE domain ZSERVICE class) that represents the application software.
6. Indicate on which operating systems the application package can run:
 - If the **Target Operating System** check box is cleared, you will not be able to select a specific operating system type. The application will be available to all end users regardless of their operating system.

- If the **Target Operating System** check box is selected, you need to select which operating systems this particular package will run on. Click, Shift-click, and Alt-click can be used to select more than one operation system.

Note

If the package was created with the Radia Packager for Windows Installer and contains a connection to the **MSIBASIC** class in the Radia Database, you have the opportunity to define the **MSI Basic Behaviors**. See the topic *Defining MSI Basic Behaviors on the New Application Wizard* on page 132.

7. If the package was created with the Radia Packager:

- Select the **Populate Windows Installer Methods** check box for Windows Installer-enabled applications. This inserts the proper calls to HP deployment and management methods for Windows Installer-enabled applications into the method variables of the service instance created by the New Application Wizard.
- If the Populate Windows Installer Methods check box is selected, select whether to use the **Default** or **Extended Verify Options**. Radia provides the ability to perform default or extended verification on Windows Installer packages.
 - ◆ Select the **Default Verify Options** to verify only the files marked as *key files* in the Windows Installer package; basically this is what MSI installer would verify. This means that if an INI File or some other type of file is not marked as a key file, and it is the only file missing from the package, it will not be repaired. Default verification sets the ZSERVICE.ZVERIFY to RadiaMSI /vd.
 - ◆ Select the **Extended Verify Options** to verify *all* files in the Windows Installer package. Therefore, if a file not marked as a key file cannot be found, it will initiate a repair. Selecting extended verification sets the ZSERVICE.ZVERIFY to RadiaMSI /ve.

The New Application Wizard will populate the following four variables in the Application class instance it creates, with the appropriate call to the Radia MSI method, as follows:

Table 5.1 ~ Variables Populated by the New Application Wizard

Variable (Option)	Inserted Method Call
ZCREATE	RadiaMSI /I
ZDELETE	RadiaMSI /u
ZUPDATE	RadiaMSI /I
ZVERIFY (Default)	RadiaMSI /vd
ZVERIFY (Extended)	RadiaMSI /ve

If the New Application Wizard displays a warning message indicating that one or more method variables are missing, update the Application class template to include the missing variables. For information on editing class templates, see *Editing a Class* beginning on page 61.

This check box should be selected only for services that are Windows Installer-enabled.

8. Click Next.

The **Application Target Type** dialog box opens.

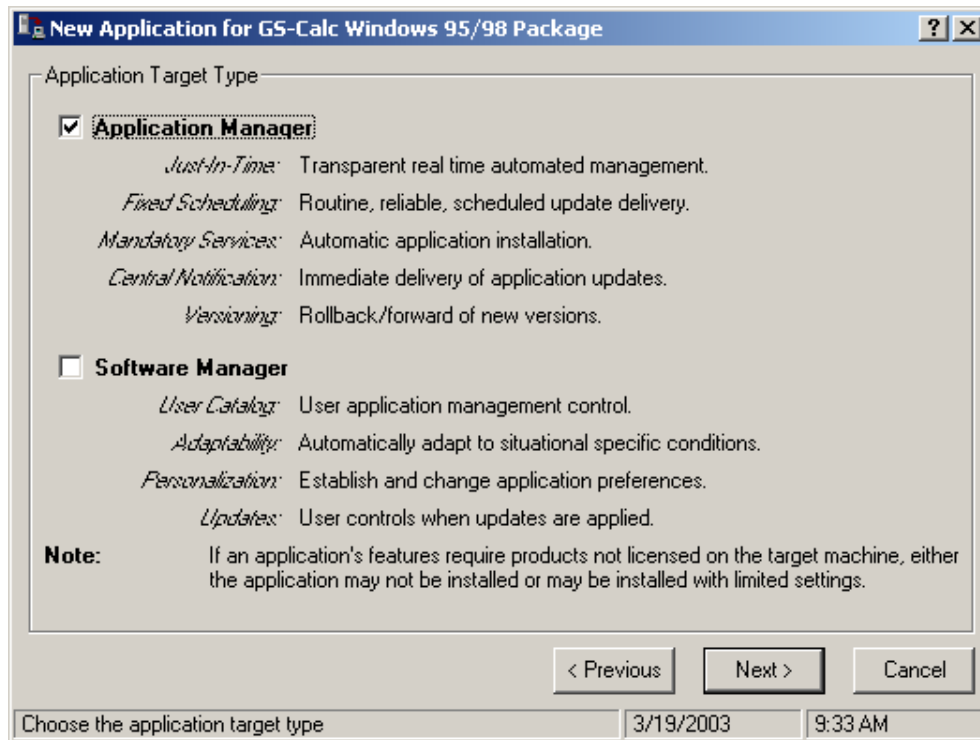


Figure 5.4 ~ Application Target Type dialog box.

Select one of the two check boxes to indicate which type of support Radia must provide for this application. The differences between the Radia Application Manager and Radia Software Manager are listed beneath each choice. You must choose one or the other. If you click **Next** without making a choice, or select both check boxes, you will receive an error message, and you will be asked to try again.

- Selecting **Application Manager** makes the application *mandatory* for end users, and stores **M** in the ZSVCMO variable of the Application (ZSERVICE) instance for this application in the Radia Database.
- Selecting **Software Manager** makes the application *optional* for end users, and stores **O** in the ZSVCMO variable of the Application (ZSERVICE) instance for this application in the Radia Database.

9. Click **Next**,

The **Application Properties** dialog box opens.

New Application for GS-Calc Windows 95/98 Package

Application Properties

Service Name: GS_CALC

Long Description: GS-CALC Spreadsheet Utility

Short Description: GS-CALC

Vendor: JPS Development

Version: 1.0

Author: JPS Development, Inc.

Web URL: www.novadigm.com

< Previous Next > Cancel

Enter the application properties 3/19/2003 9:36 AM

Figure 5.5 ~ Application Properties dialog box with application properties entered in the appropriate fields.

Fill in the data entry fields as described in Table 5.2 below.

Table 5.2 ~ Application Properties Data Entry Fields

Property	Stored in ZSERVICE Attributes	Usage
Long Description	DESCRIPT	This name will appear to subscribers in the Radia Subscriber Service List.

Table 5.2 ~ Application Properties Data Entry Fields

Property	Stored in ZSERVICE Attributes	Usage
Short Description	ZSVCNAME NAME	This will appear in the Radia System Explorer's tree view as the friendly name of the Application (ZSERVICE) instance created by the New Application Wizard.
Vendor	VENDOR	The subscriber can configure his user interface to view the vendor name in his Radia Subscriber Service List. This will help him identify applications.
Version	VERSION	The subscriber can configure his user interface to view the version number of the application in his Radia Subscriber Service List.
Author	AUTHOR	The subscriber can configure his user interface to view the author of the application in his Radia Subscriber Service List.
Web URL	URL	The subscriber can configure his user interface to view the Web URL of the application in his Radia Subscriber Service List. This is the Web address where he can obtain additional information about the application.

When completed it looks like Figure 5.5 on page 127.

10. Click **Next**.

The **Application Level Event Reporting** dialog box opens.

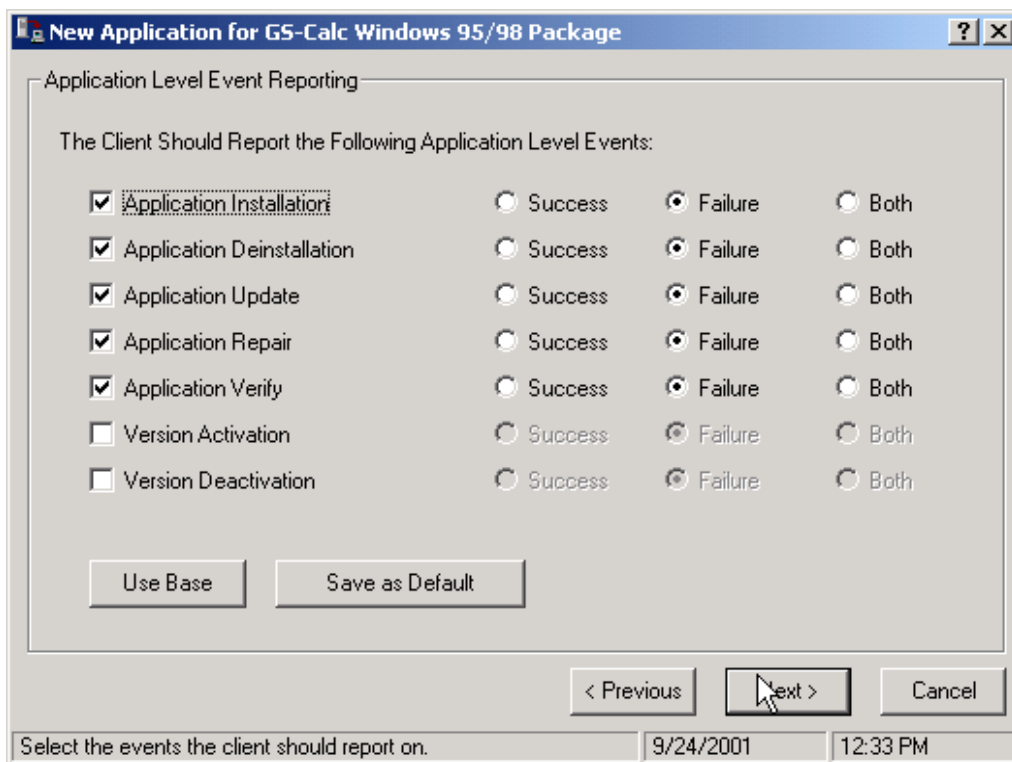


Figure 5.6 ~ Application Level Event Reporting dialog box for reporting application level events to the Radia Configuration Server.

Use this dialog box to indicate which events the Radia Client must report to the Radia Configuration Server. To enable reporting for an event, select its check box. Once reporting is enabled for an event, indicate whether to report success, failure, or both by selecting the event's associated option buttons. The example in Figure 5.6 requires Radia Clients to report failure of application installation, de-installation, update, repair, or verify to the Radia Configuration Server.

There are also two buttons: **Use Base** and **Save as Default**.

- Select **Use Base** if you would like the package to inherit the settings from the base instance of the Application (ZSERVICE) instance.
- Select **Save as Default** to use the current settings as the default settings for the **Application Level Event Reporting** dialog box.

11. Click Next.

The **Application Summary** dialog box opens, confirming your selections and data entry in the **New Application Wizard**.

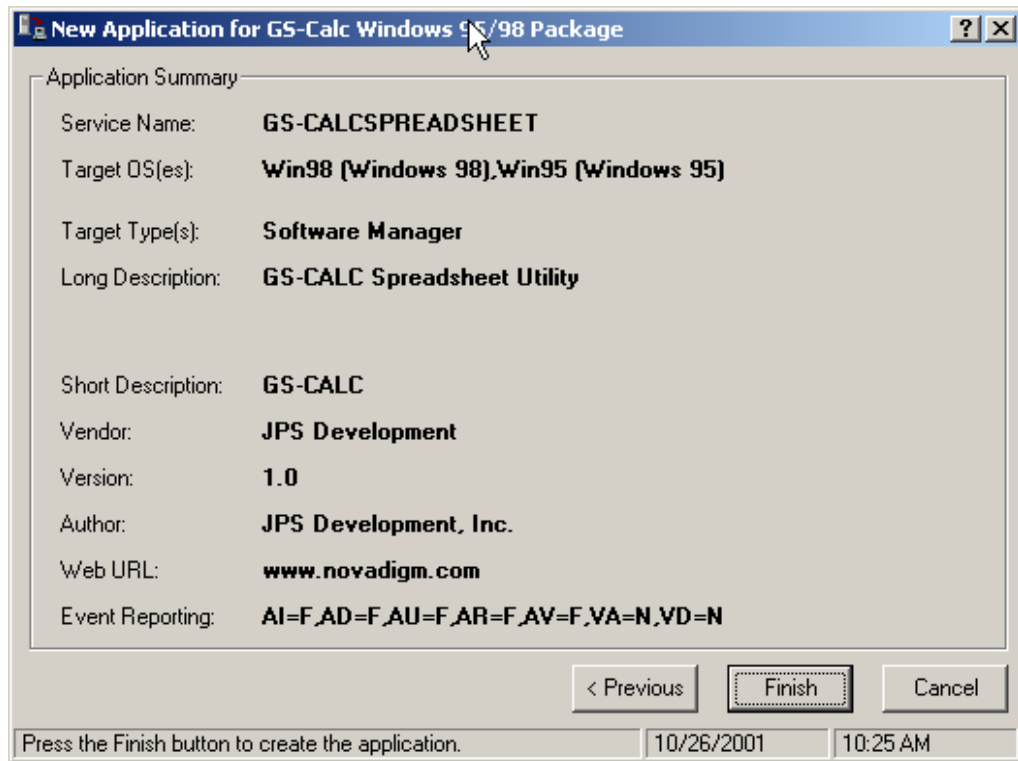


Figure 5.7 ~ The Application Summary dialog box summarizes and confirms selections made in the New Application Wizard.

The **Event Reporting** summary at the bottom of this dialog box is an abbreviated specification that summarizes the options that were selected for event reporting in the **Application Level Event Reporting** dialog box. It consists of a series of keyword=value pairs, separated by commas. Each keyword identifies a reporting event as described in Table 5.3 on page 131.

Table 5.3 ~ Event Reporting Keywords and their Denotations

Keyword	Reporting Event
AI	Application Installation
AD	Application Deinstallation
AU	Application Update
AR	Application Repair
AV	Application Verification
VA	Version Activation
VD	Version Deactivation

The value associated with each keyword indicates which option button, if any, was selected for the event, as follows:

Table 5.4 ~ Values Associated with each Event Reporting Keyword

Value	Selected Option Button
S	Success
F	Failure
B	Both (report success and failure)
N	None (no reporting for the event)

- 12.** If you need to make changes, click **Previous** to return to the appropriate dialog box and make your changes.

Otherwise, click **Finish** to create the service instance for the application and link it to the promoted package. When you do so, the following message opens.

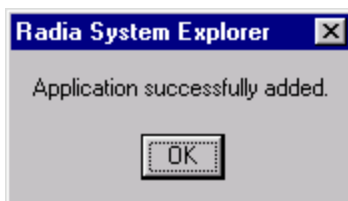


Figure 5.8 ~ Application successfully added confirmation box.

The GS-Calc Windows 95/98 package is now a ZSERVICE. The ZSERVICE is now ready to be connected to users for deployment.

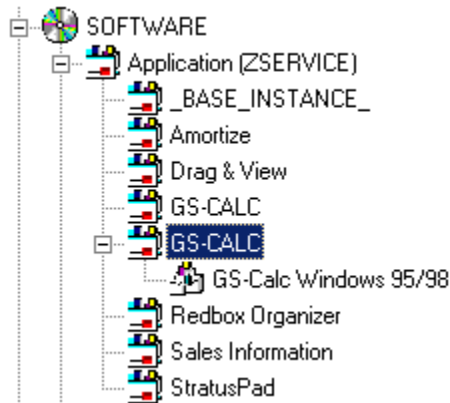


Figure 5.9 ~ The highlighted GS-Calc package is the service you just created.

Defining MSI Basic Behaviors on the New Application Wizard

In contrast to Advanced Package Management, where Radia 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 simply to be installed and uninstalled by Radia, but whose components are not to be managed by Radia. That is, the Radia client 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 Radia Packager for Windows Installer. These packages have a connection to the **MSIBASIC** class in the Radia Database.

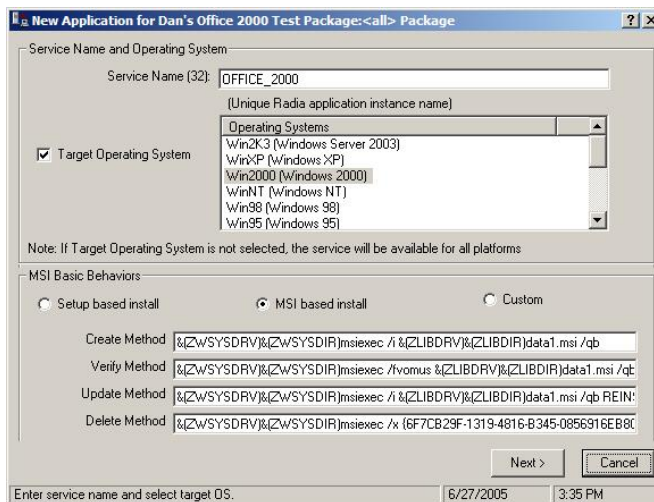


Figure 5.10 ~ New Application Wizard with MSI Basic Behaviors section.

Packages that cannot be administratively installed will default to Basic Package Management with the Radia manageability features disabled. Use the **MSI Basic Behaviors** section of the **New Installation Wizard** dialog 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 for you. 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 may add them to the **Create Method**. Not all MSI applications support the same command line switches. For setup.exe, please refer to your application's documentation for details.
 - **MSI based install**
The package installs with MSIEXEC.
 - **Custom**
The package uses a custom install method.
2. Review and modify, if necessary, the native MSI commands and switches 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 Radia System Explorer when the Radia Application Manager is installed. Use the Version Group Editor to create or edit instances of the Version Group class. Features of the Version Group Editor enables you to create, edit, and delete Version instances of the Version Group and control the Version Group's deployment.

Creating a Version Group Instance

The example in this section describes how to create an instance of the Version Group class that represents a set of versions of an application.

To create a Version Group instance for a package

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box appears.

Note

The **User ID**, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**.
3. The Radia System Explorer splash screen appears briefly.
4. Double-click **PRIMARY**.
5. Expand the **SOFTWARE** domain.
6. Right-click **Version Groups (VGROUP)** and select **New Instance** from the shortcut menu.

A **Create Instance** dialog box opens.

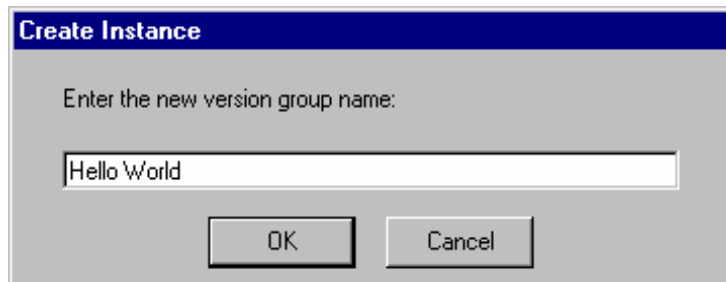


Figure 5.11 ~ Create Instance dialog box.

7. Type a friendly name for the new Version Group, for example, Hello World.
The friendly name will appear in the tree view of Radia System Explorer.
8. Click **OK**.

The **Editing Version Group** dialog box opens.

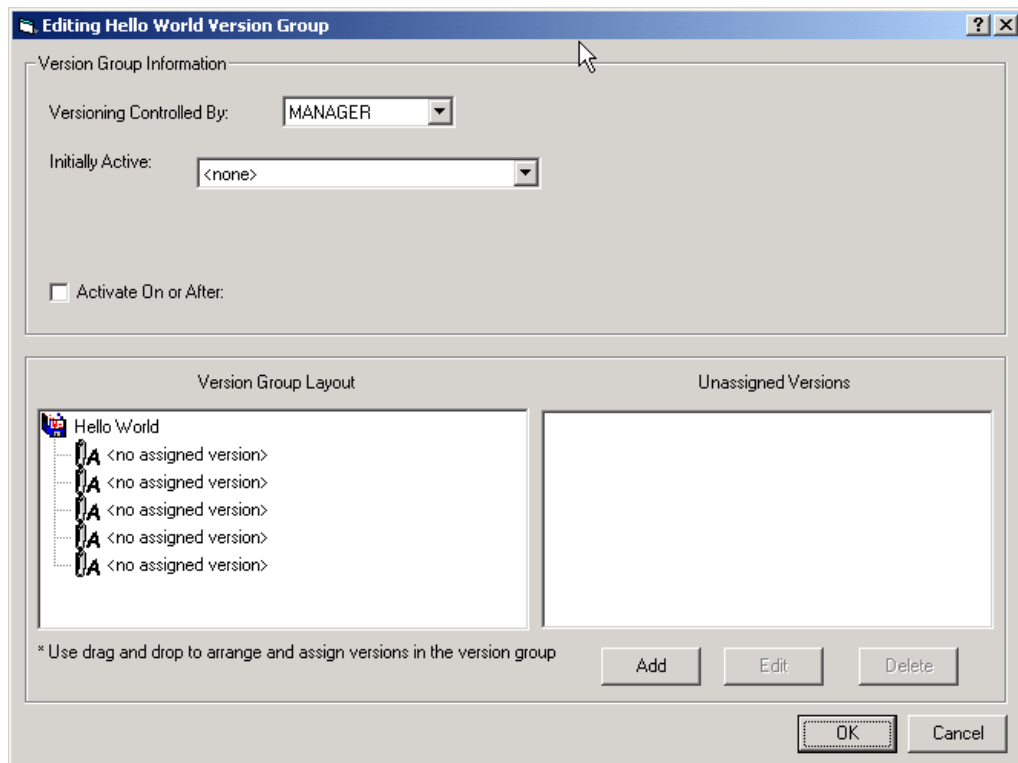


Figure 5.12 ~ Editing Version Group dialog box.

The **Version Group Editor** contains all the controls you need to manage the content and deployment of a Version Group.

9. Make selections on this dialog box based on the following explanations.
 - **Versioning Controlled By** drop-down list
This control determines whether the Radia administrator (**Manager**) or end user (**Client**) chooses the version to deploy.
 - ◆ Choose **MANAGER** to select the Radia Administrator. The Radia Application Manager supports version control by the Radia administrator only, it does not support version control by the end user.
 - ◆ Choose **CLIENT** to select the end user. Setting the control to **CLIENT** will disable version deployment.

- **Initially Active** drop-down list

This control selects which version to activate on the client computer the next time the subscriber's Radia Client connects to the Radia Configuration Server. The available versions are those that appear in the Version Group Layout section of the Editing Version Group dialog box.

- **Activate On or After** check box

This control provides access to additional controls that the Radia administrator can use to prevent version activation prior to a specific date and time. When selected, the controls appear as follows:

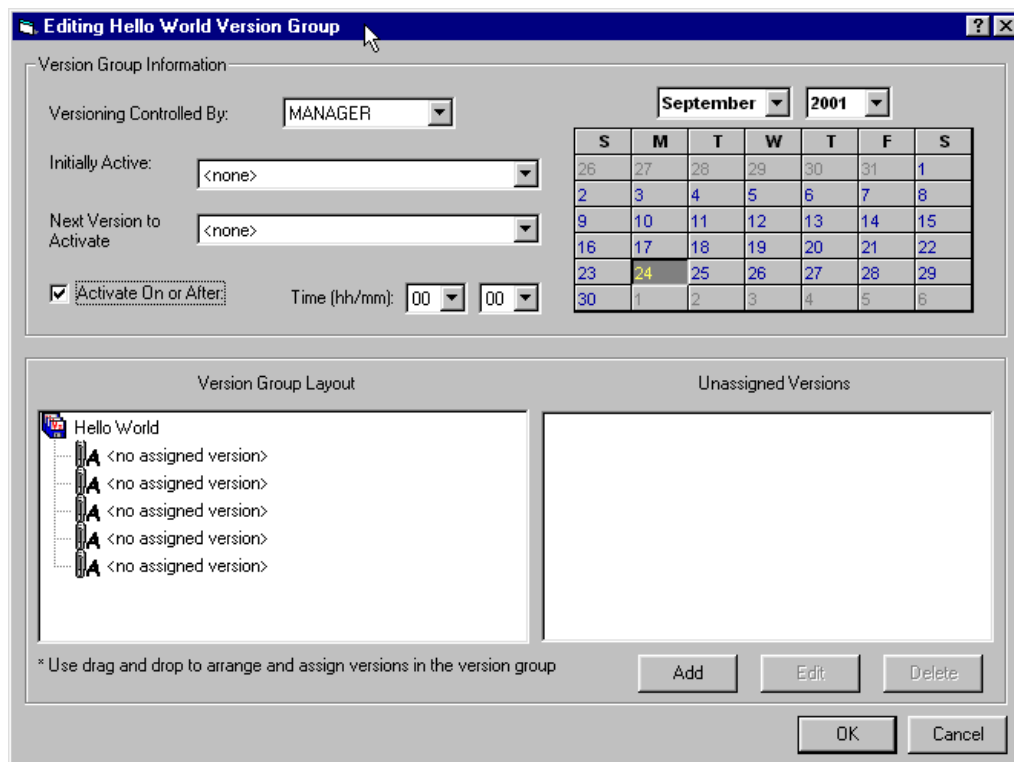


Figure 5.13 ~ The Editing Version Group dialog box displays time and date controls when the **Activate On or After** check box is selected.

Additional features are displayed when selecting the **Activate On or After** check box.

- **Next Version to Activate**

Use the drop-down list to indicate which version of the application to activate if the newly deployed version fails to install and activate successfully.

- **Calendar and Time**

These controls set the date and time used for deployment. Use these functions to set the Timer.

Note

If you will be using the Version Group Editor to maintain a TIMER instance, there are a few things to consider:

- Timers are only created and connected to a VGROUP if versioning is controlled by the Radia Configuration Server. This is done by selecting MANAGER on the Versioning Controlled By drop-down list in the Version Group Editor. In addition, with MANAGER selected, the Activate On or After checkbox must be selected. By selecting this checkbox, the Calendar and Time controls that the administrator sets to deploy the next version will become available.
- The calendar will not allow the administrator to select a date earlier than the current day.
- For more information about Timers, see the *Radia Application Manager Guide*.

Use the time and calendar controls to indicate the date and time when the designated (i.e., **Initially Active**) version should first become available for deployment.

The **Version Group Layout** and associated controls at the bottom of the window enable you to create and edit Version instances and assign them to the Version Group as needed.

10. Click **Add** at the bottom of this dialog box.

A **Create Version** dialog box opens.

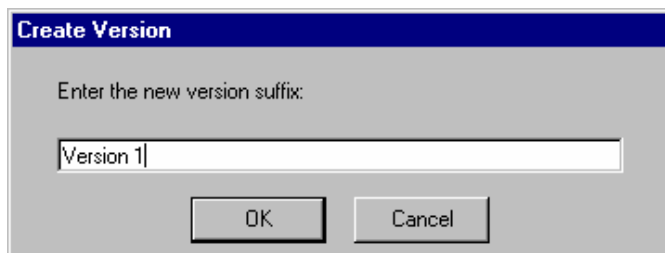


Figure 5.14 ~ Create Version dialog box.

11. Enter a suffix for the friendly name for the new Version instance in the text box.

The friendly name for the Version instance will be a concatenation of the Version Group's friendly name and the suffix you enter here. In the example presented here, the friendly name assigned by the Radia System Explorer for this Version instance will be "Hello World Version 1".

12. Click **OK**.

The **Version Editor** dialog box opens.

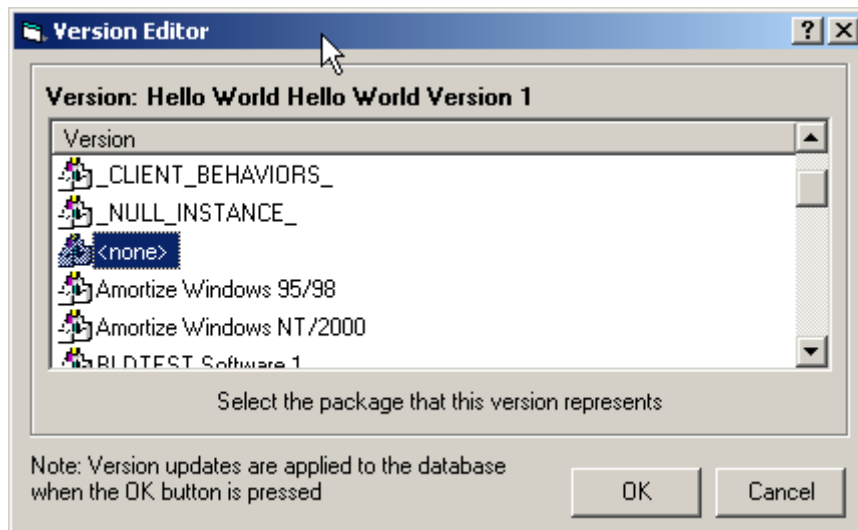


Figure 5.15 ~ Version Editor dialog box.

The Version Editor dialog box contains a list of Application Package instances currently stored in the Radia Database. Use this dialog box to associate the newly created Version instance with an Application Package instance.

13. Scroll down the list until the desired Application Package instance appears in the window. In this example, we will connect it to the Amortize Windows 95/98 package instance. Select **Amortize Windows 95/98** by double-clicking it, or single-clicking it and clicking **OK**.

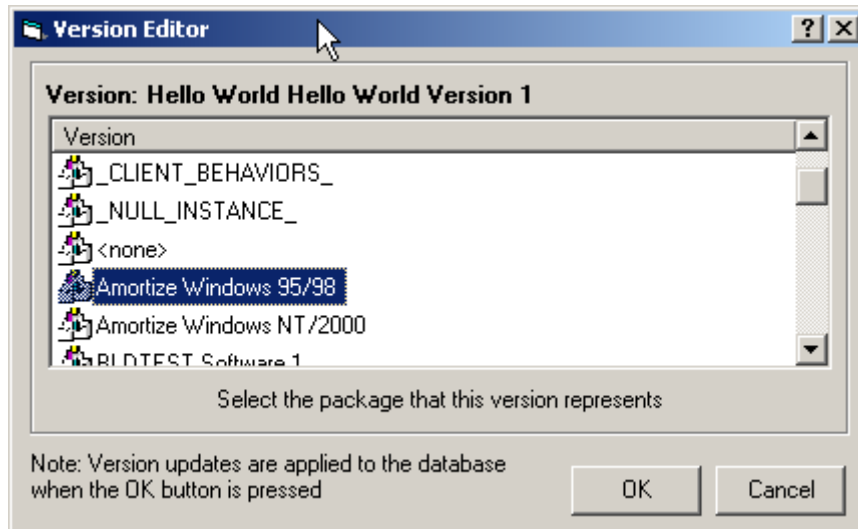


Figure 5.16 ~ Connect the Amortize Windows 95/98 package to the Hello World Version Group.

The Version instance will be created, you will be returned to the **Editing Version Group** dialog box, and the version instance will appear in the **Unassigned Versions** list.

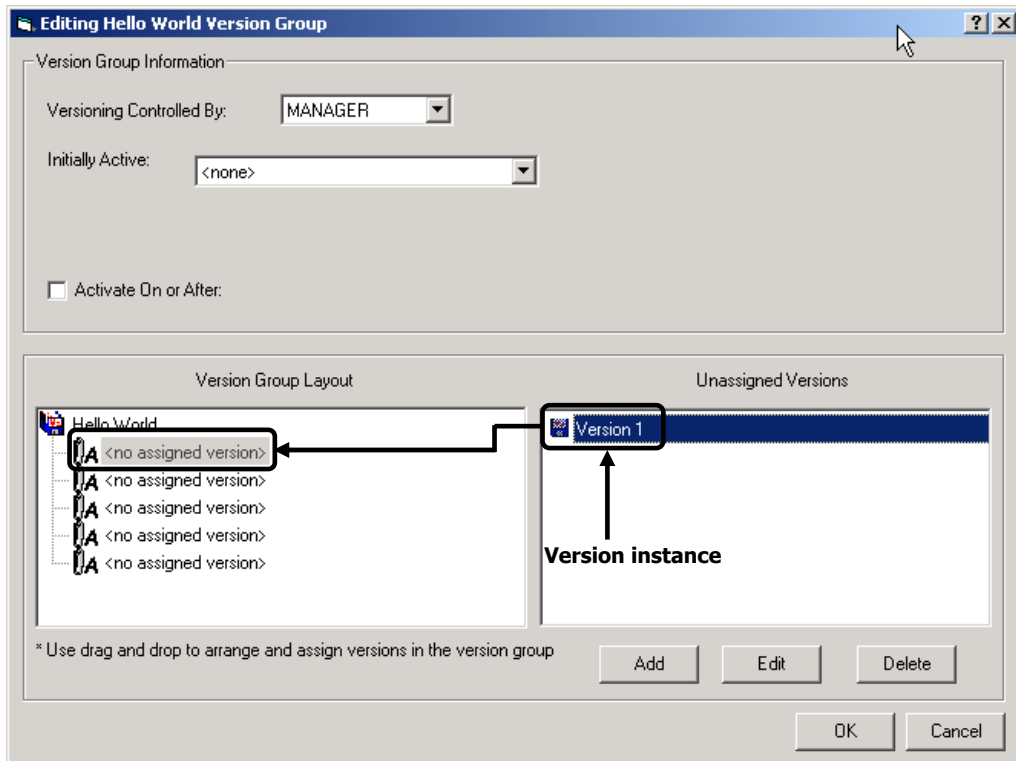


Figure 5.17 ~ A Version instance created and listed in the Unassigned Versions list.

14. Repeat the process of adding version instances for each version of the application software to be made available to end users via this Version Group.

Version Group Assignments

The previous exercise showed how to create a version instance for a package. Once the version is created, you will need to connect it to a Version Group.

To assign a Version Instance to a Version Group

1. Drag a Version instance icon from the **Unassigned Versions** list to one of the connections labeled **<no assigned version>** in the **Version Group Layout**.
2. Repeat this until all needed versions have been assigned.

The **Editing Version Group** window will look like Figure 5.18 on page 142:

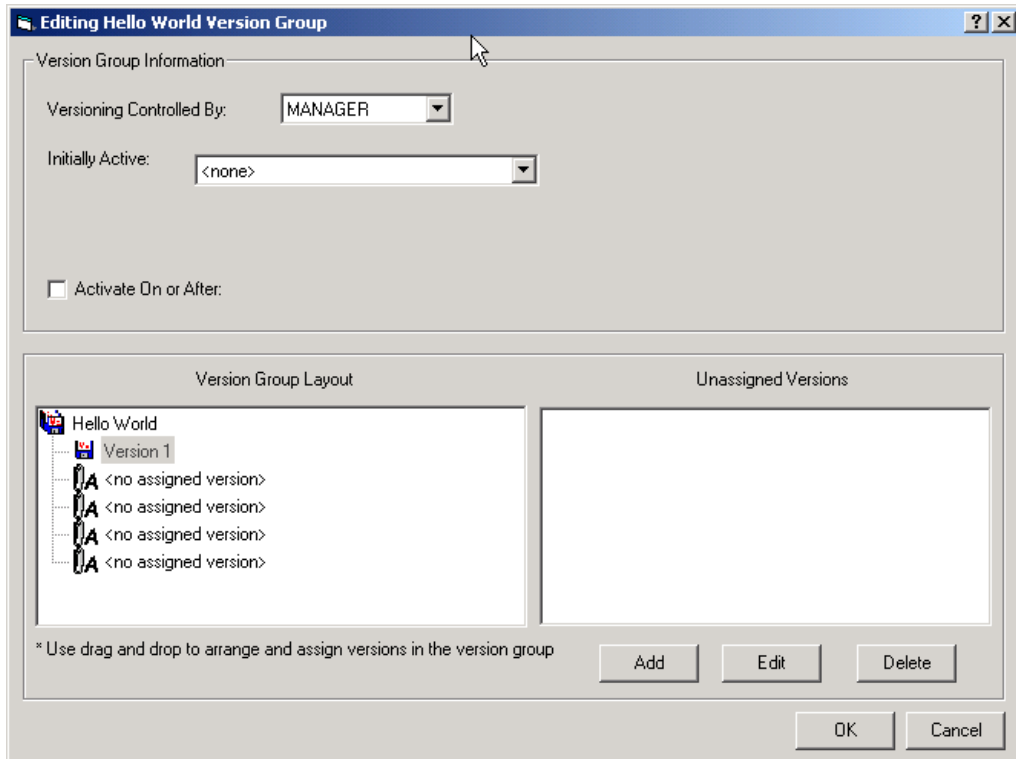


Figure 5.18 ~ Editing Version Group dialog box showing Version1 assigned to Hello World.

3. Click the **Initially Active** drop-down arrow to display assigned versions. You may choose to deploy any of the assigned versions, or **<none>**.

When a Radia Client connects to the Radia Configuration Server, the packages (including files, links, shortcuts, and registry entries) associated with all assigned versions are transferred (if they are not already present on the client computer), in compressed form, to holding locations on the client computer. The version you select in the **Initially Active** drop-down list is then installed and activated on the client computer.

Select the version to deploy from the Initially Active drop-down list. It will be highlighted in the Version Group Layout list (see Figure 5.19 on page 143).

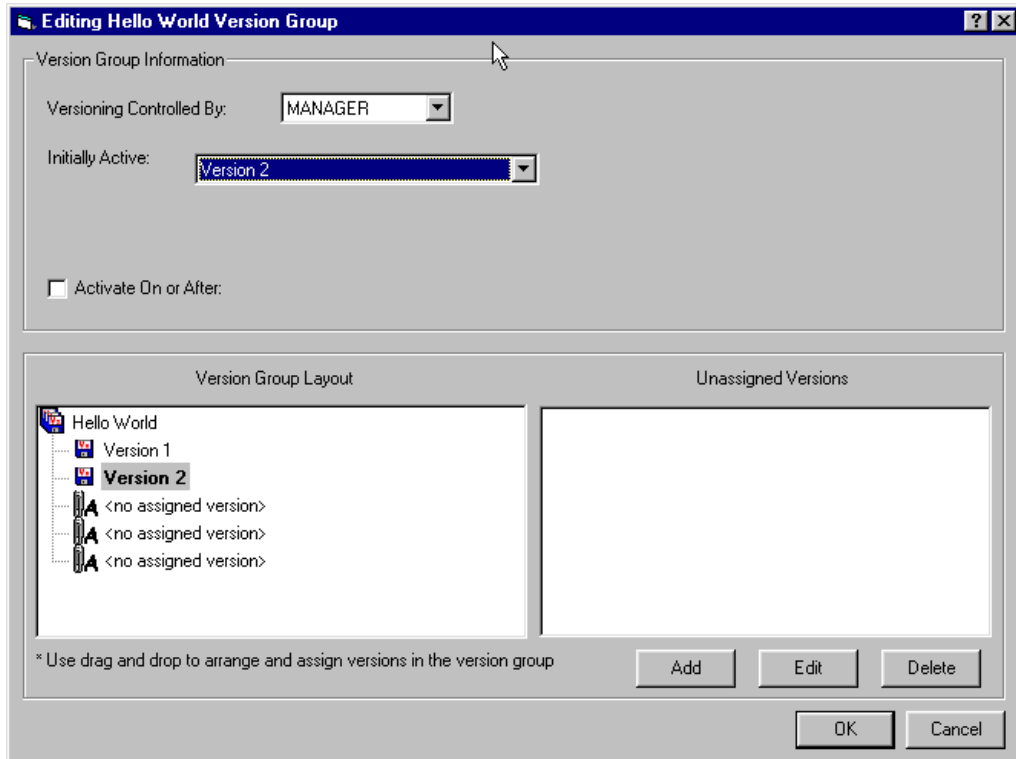


Figure 5.19 ~ The selected version for deployment, Version 2, is highlighted in the Version Group Layout list.

If the **Initially Active** control is set to **<none>**, the Radia Client connection to the Radia Configuration Server will transfer all assigned packages that have not yet been transferred, but none will actually be installed on the client computer. This could be quite confusing for the end user, given the following scenario: Your end user visits your organization's Radia Web page (using his Web browser with the Radia plug-in installed). A client connection takes place and the end user chooses to install a versioned application from the Service List. On the Radia Software Manager user interface, the user will see the file download activity as the version packages are transferred in compressed form and stored in holding locations on his computer. The Service List will indicate that the application has been installed, but since the **Initially Active** control indicates **<none>**, no version will be activated. Again, this situation can be disconcerting to end users and result in unnecessary calls to your organization's support group.

You can avoid this situation with proper planning. If you need to deploy a versioned application without activating any version immediately, consider making one of the versions a stub application that simply displays a suitable message to the end user. For example, create

a stub application whose job is to simply indicate that the application is not yet ready for installation. Instead of selecting **<none>** for the **Initially Active** version, select the stub application as the initially deployed version until you are ready to deploy the real thing.

To remove a Version Assignment in a Version Group

- Drag-and-drop the version icon from the **Version Group Layout** to the **Unassigned Versions** list. The assignment will be removed.

To delete a Version Instance from a Version Group

1. Highlight the desired Version instance, and click **Delete**. The Version instance will move to the Unassigned Versions list (if it is not already there), and its icon will appear dimmed to indicate that it has been deleted.

Deleted Version instances listed in the **Unassigned Versions** list will not actually be deleted from the Version Group until you close the Version Group Editor. While the Version Group Editor is still open, you can restore a deleted Version instance by clicking it, and then clicking **Un-delete**.

2. Once the Version Group has been completely configured, click **OK** to close the window.

To edit a Version Group

1. Right-click on the **Version Group** instance in the Radia System Explorer.

A shortcut menu opens.

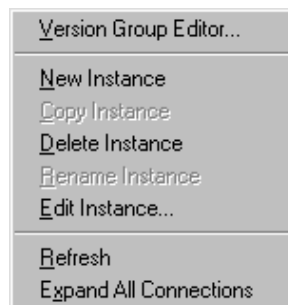


Figure 5.20 ~ Version Group class shortcut menu.

2. Select **Version Group Editor** from the menu, and continue as described above.

Service Optimization

The intent of Service Optimization is to maximize efficiency of distributing periodic updates to Radia packages comprising a Radia service. Later packages contain revisions and additions to files published in earlier packages.

Service Optimization uses **byte level differencing** to create patches.

■ Byte Level Differencing

Byte Level Differencing is the process of comparing two sets of data, at the byte level, and then identifying their differences. The differences between the two packages are the bytes that are required to upgrade or fix an application.

■ Patch

A patch is a published package of data (bytes) that is required to upgrade or fix an application that is already published and installed on a client 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.

Byte level differencing is supported in the Radia Packager, the Radia Database, and the Radia System Explorer.

■ The Radia Packager

The Radia Packager calculates the *message digest algorithm (MD5)* signature of the file as it is being promoted. 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 Radia Configuration Server. For more information about MD5 signature, see the following Web address:

<http://www.faqs.org/rfcs/rfc1321.html>

■ The Radia Database

The PATCH domain contains the list of available patches in a new PATCH class. The file signatures of the patch are used to identify the associated instance that contains the data for the patch. For more information about the PATCH domain, see the *Radia Configuration Server Guide*.

Note

Prior to deploying patches to your user's, the ZOBJPTCH variable in the USER class in the POLICY domain must be changed from N to Y.

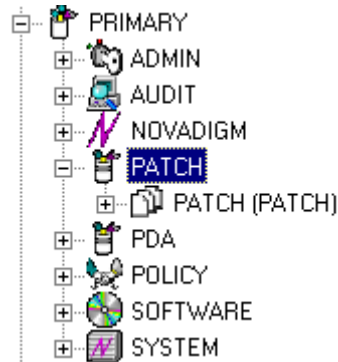


Figure 5.21 ~ PATCH.CLASS in the PRIMARY file.

The following table describes the attributes in the PATCH class.

Table 5.5 ~ Attributes in the Patch Class

Attribute Name	Description
NAME	Friendly Name – the value of ZRSCCFIL attribute in the To instance.
STATUS	Status of the patch. Valid values are: Queued , Rejected , or Built .
MESSAGE	Contains the explanation of the status. Reference this attribute to determine if a patch was built successfully.
PVERSION	The version of the patch builder.
POS	The operating system on which the patch was built.
ZRSCDATE	The date when the patch was built.
ZRSCTIME	The time when the patch was built.
ZRSCSIZE	The size of the patch.
ZCMPSIZE	The compressed size of the patch.
ZRSCSIG	The content-based signature of the patch type.
ZRSCTYPE	Type of resource. The only valid value is PATCH.
ZRSCCSTA	Status of patch resource. The only valid value is 999.
	The controls used in patch generation:
ZCONTRL1	<ul style="list-style-type: none"> ZSTOPPCT is the percentage of target data when the patch will stop building the path.
ZCONTRL2	<ul style="list-style-type: none"> ZSTOPMAX is the maximum size of the patch.
ZCONTRL3	<ul style="list-style-type: none"> ZTIMEOUT is the time allotted to patch creation before aborting the Create.

Table 5.5 ~ Attributes in the Patch Class

Attribute Name	Description
	See Table 5.7 on page 159.
SIGTYPE	For now, MD5 is the only signature type supported.
FROMSIG	The content-based signature of the From file.
FROMSIZE	The size of the From file. This value is taken from the ZRSCSIZE of the From instance.
FROMFDCI	Original location of the From instance
FROMOID	Object ID of the original From instance.
TOSIG	The content-based signature of the To file.
TOSIZE	The size of the From file. This value is taken from the ZRSCSIZE of the To instance.
TOFDCI	The original location of the To instance.
TOOID	The object ID of the original To instance.

■ The Radia System Explorer

Optimize Service is an option that has been added to 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 will first need to promote your packages using the Radia Packager. For instructions on how to do this, see the *Radia Application Manager Guide, Publishing Applications and Content* chapter.

Note

- Patches can only be created for components that contain a signature. Initially, only MD5 is supported.
- Components 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.

Caution

Because the Radia Packager now calculates and uses the MD5 signature, and this signature is vital to the patching functionality, there is no backwards compatibility to previously promoted components.

This section will provide you with detailed instructions for:

- Creating an Application (ZSERVICE) in the SOFTWARE domain for those packages for which you will create a patch.
- Creating the patches using the Optimize Service wizard.

We have provided you with detailed information about each of the screens that you will encounter.

In this example, you will optimize a series of text files called BLD. These files were packaged and promoted to the Radia Configuration Server using the Radia Packager. Each text file is its own PACKAGE in the SOFTWARE domain.

Once you've promoted your packages you will need to create a ZSERVICE instance.

Caution

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.

Note

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.

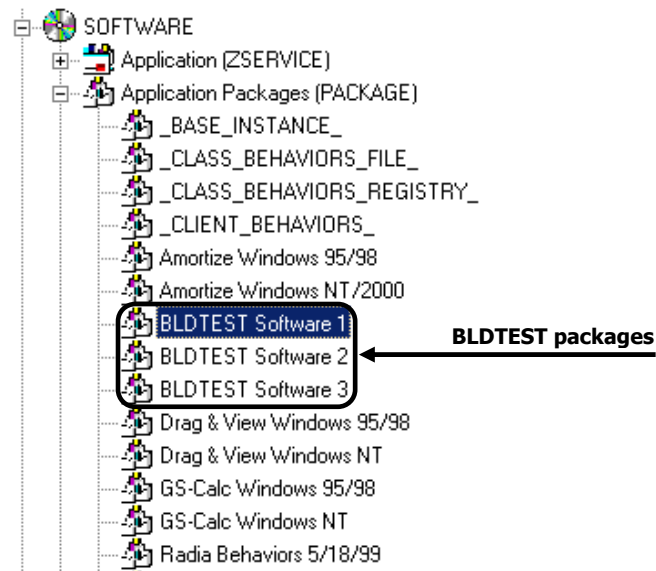


Figure 5.22 ~ BLDTEST packages in the SOFTWARE domain.

2. Right-click on the **Application (ZSERVICE)** class.

A shortcut menu opens.

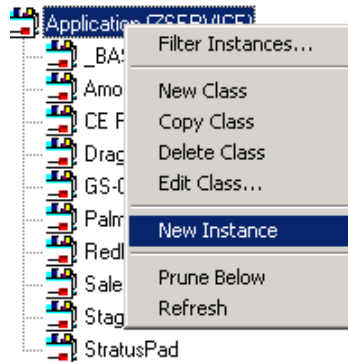


Figure 5.23 ~ ZSERVICE shortcut menu.

3. From the shortcut menu, select **New Instance**.
 4. Create a new instance named **BLD_CREATE**. Click **OK**. The BLD_CREATE service is added to the Radia Database, as shown in Figure 5.24 below.
-

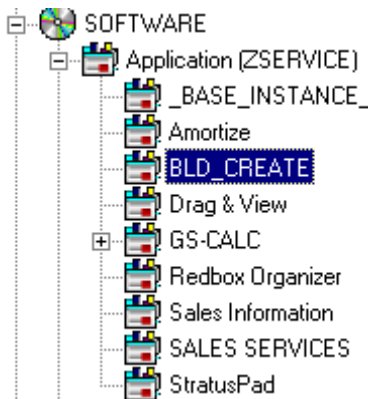


Figure 5.24 ~ BLD_CREATE ZSERVICE instance.

5. Drag-and-drop the BLDTEST1 package to the BLD_CREATE instance in the Application (ZSERVICE) class.
 6. Drag-and-drop the BLDTEST2 package to the BLD_CREATE instance in the Application (ZSERVICE) class.
-

7. Drag-and-drop the BLDTEST3 package to the BLD_CREATE instance in the Application (ZSERVICE) class.

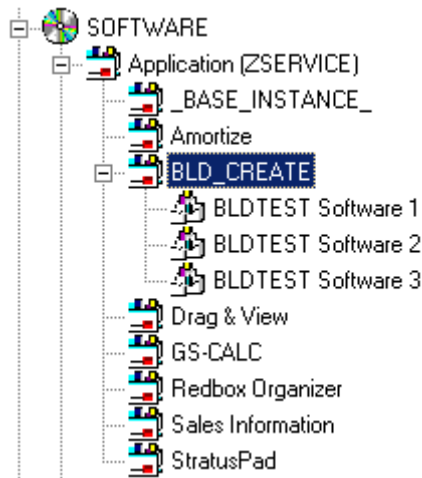


Figure 5.25 ~ BLD_CREATE ZSERVICE instances.

Once you've created the Application (ZSERVICE) containing the Application Packages (PACKAGE) you can begin to create your patches.

Just as you create a plan or policy for users, you may 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 to document the packages that will create your patches. When building different patches within 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.

8. Right-click on the Application (ZSERVICE) class **BLD_CREATE**. A shortcut menu opens.

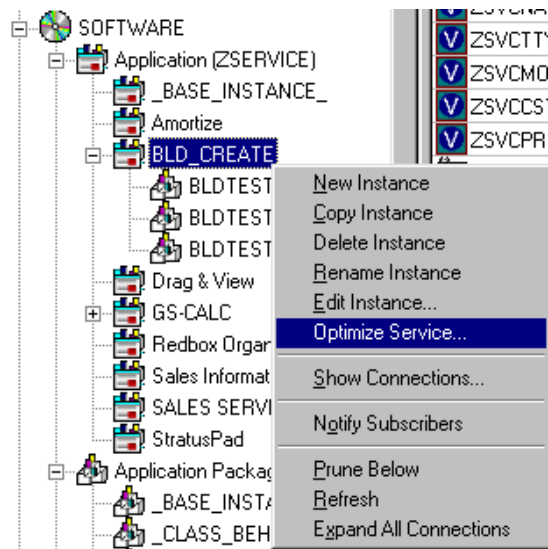


Figure 5.26 ~ Optimize Service from the BLD_CREATE ZSERVICE.

9. Select Optimize Service.

The Service Optimization Wizard opens.

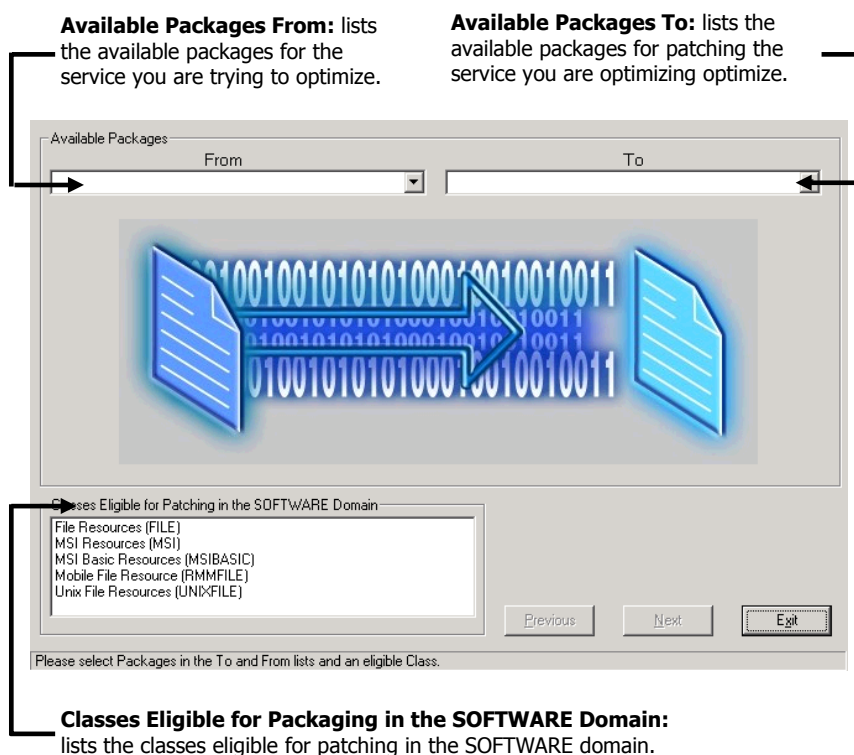


Figure 5.27 ~ Service Optimization Wizard—Available Packages dialog box.

There are three areas in this first screen of the Service Optimization wizard:

- Available Packages From**
 This list displays the available packages for the service you are trying to optimize. When you click on the **From** drop-down arrow, all of the packages for the Application (ZSERVICE) will be displayed. In this example, clicking on the **From** list will show all of the packages in the BLD_CREATE Application (ZSERVICE).
- Available Packages To**
 Once you've selected the **From** package, the **To** list becomes populated with those packages available for patching in the service you are optimizing.
- Classes Eligible for Patching in the SOFTWARE Domain**
 Select the appropriate class eligible for patching within the SOFTWARE domain.

Available classes are File Resources (FILE), MSI Resources (MSI), and/or Unix File Resources.

10. Click on the **Available Packages From** drop-down arrow.

The packages that are available for the service you selected in steps 5-7 (on page 150), BLD_CREATE, are listed.

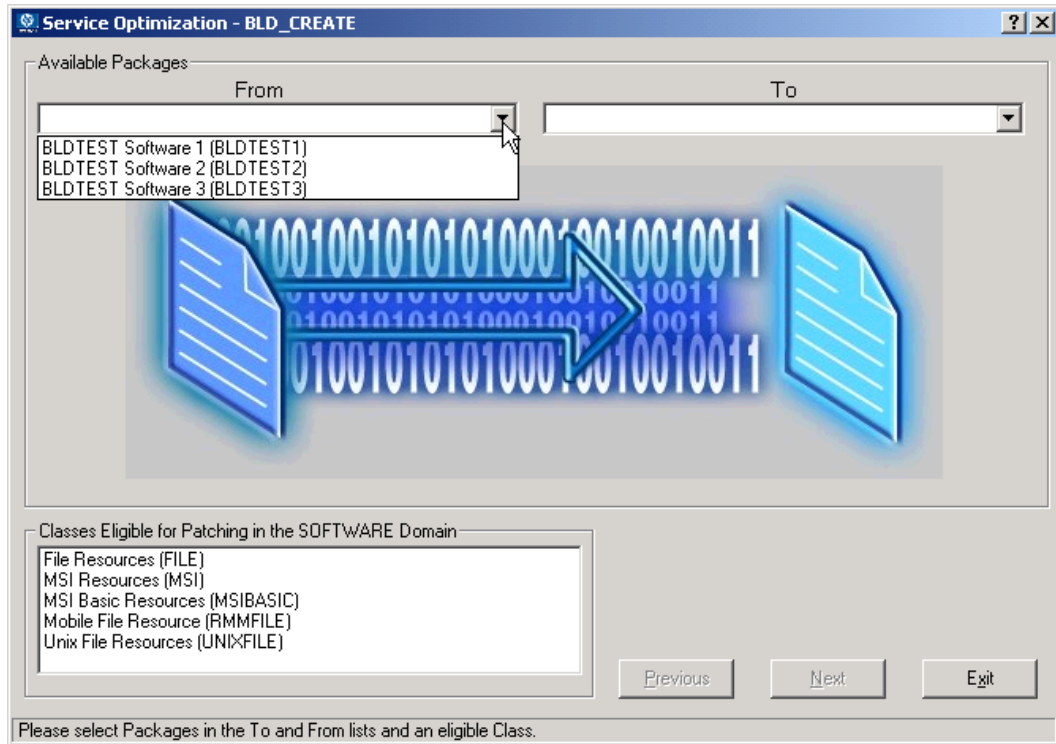


Figure 5.28 ~ Select which package you would like to optimize.

11. From the **Available Packages From** drop-down list select **BLDTEST Software 1 (PACKAGE 1)**.

The **Available Packages To** list is populated with the packages that are available for patching in the BLD_CREATE service.

12. From the **Available Packages To** drop-down list select **BLDTEST Software 2 (SOFTWARE 2)**.

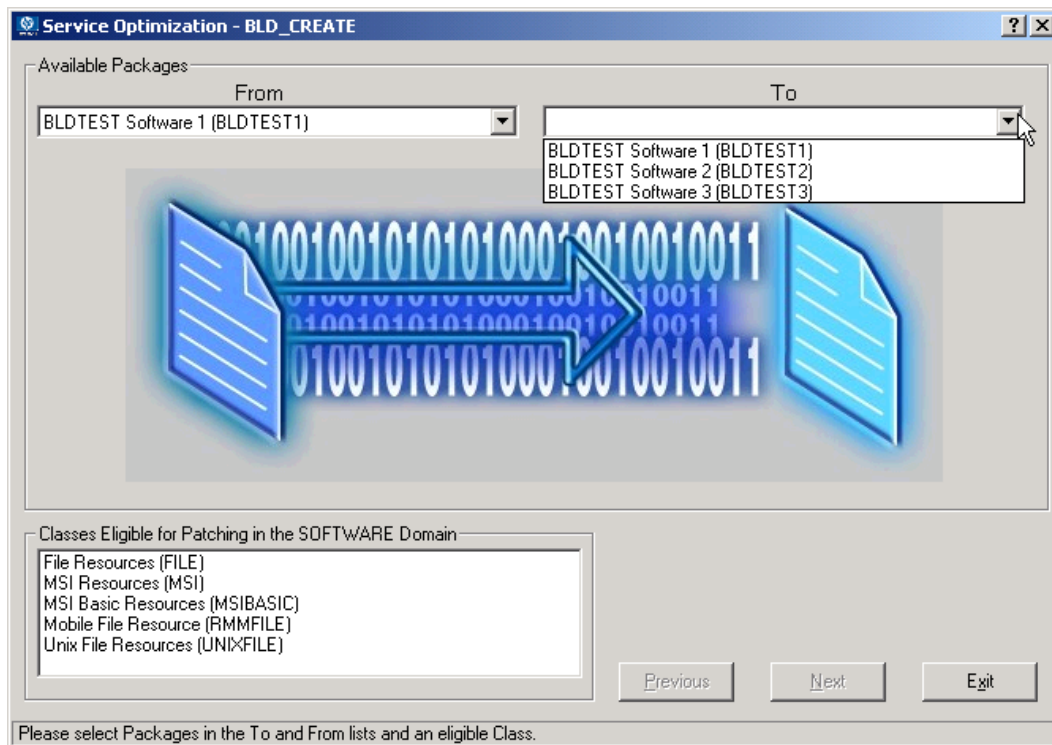


Figure 5.29 ~ Select the package you would like to compare the "from" against.

13. From the **Classes Eligible for Packaging in the SOFTWARE Domain** select the appropriate class for patching in the SOFTWARE domain. Choose the class where the file resources for the package are located.

Note

Currently, File Resources (FILE), MSI Resources (MSI) and Unix File Resources (UNIXFILE) qualify for patch building.

In this example, the only class available to be patched is the **File Resource (FILE)** class.

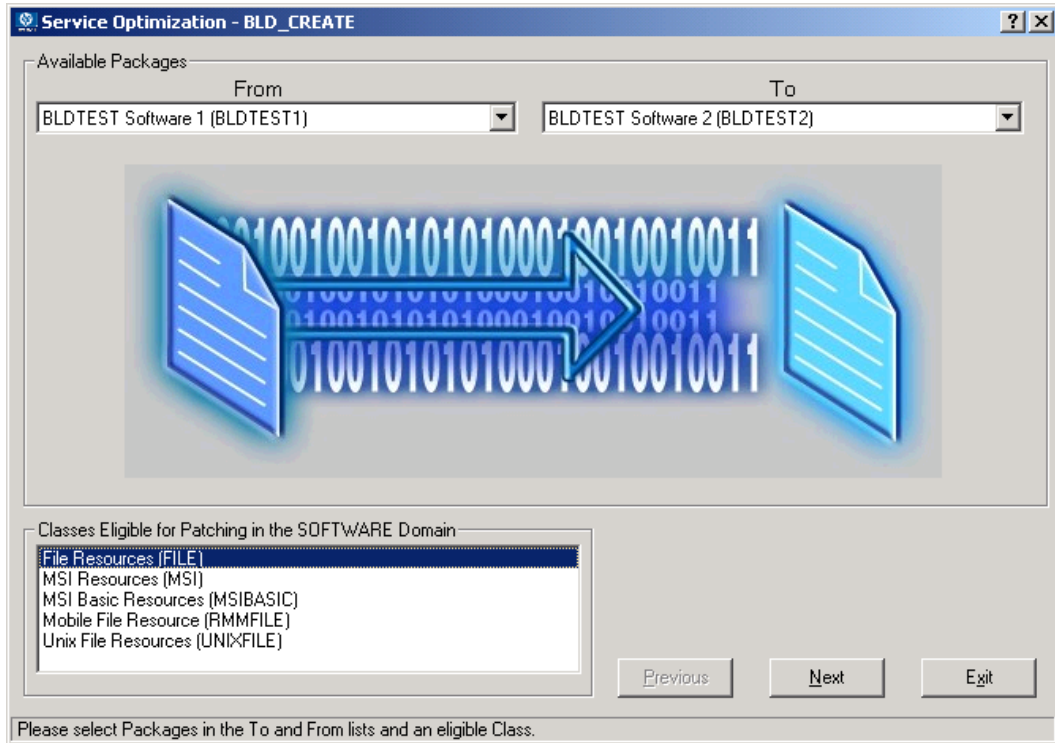


Figure 5.30 ~ Completed Service Optimization Wizard.

Note

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.

14. Click **Next** to continue.

The **Files and Progress** dialog box opens.

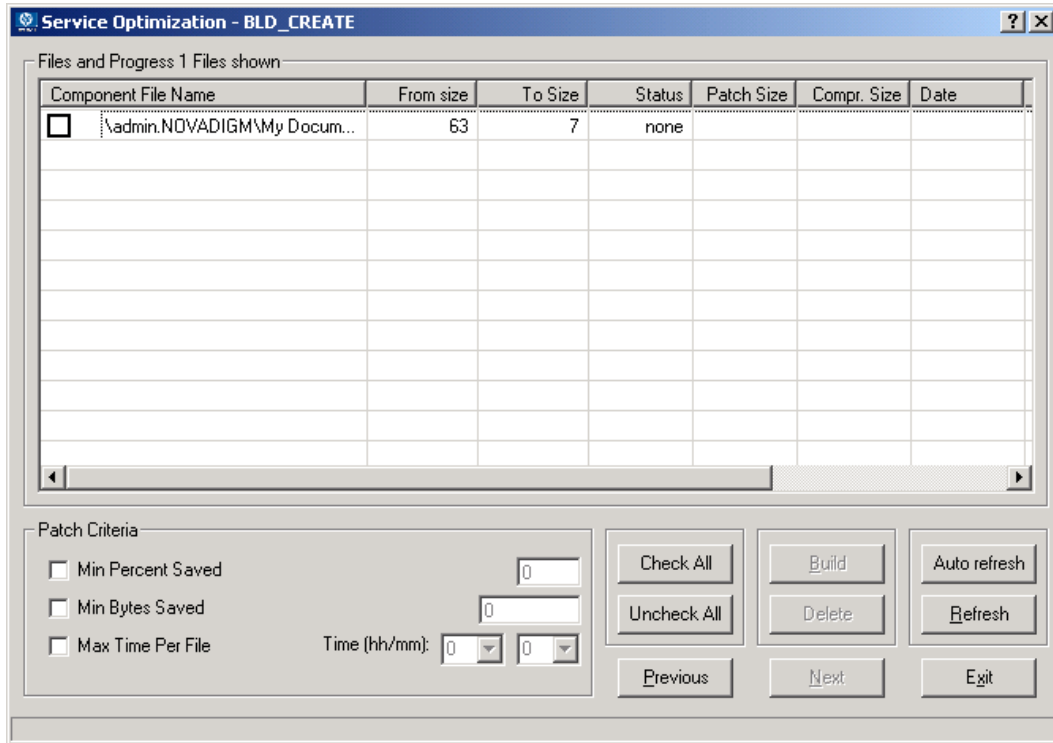


Figure 5.31 ~Files and Progress dialog box.

This dialog box displays files and progress information. The Service Optimization wizard gathers the instance information based upon the selections made in the **Available Packages** dialog box.

The information in the **Files and Progress** dialog box is described in Table 5.6 on page 158.

Table 5.6 ~ Files and Progress Information

Column	Description
Component File Name	The name of the patch being created.
From Size	The original resource package size.
To Size	The original target resource size.
Status	The status of the package. This can be none, queued, rejected, active, or built.
Patch Size	The comparison 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.
Date	The date the package was published and promoted.
Size Limit	This information is pulled from the Patch Criteria, Min Percent Saved check box and field. This field will become populated once the Patch has been built or upon termination of the patch building process.
% Limit	This information is pulled from the Patch Criteria, Min Bytes Saved check box and field. This field will become populated once the Patch has been built or upon termination of the patch building process.
Time Limit	This information is pulled from the Patch Criteria, Max Time Per File check box and field. This field will become populated once the Patch has been built or upon termination of the patch building process.

It is possible to place patch-building restrictions, such as percent size, bytes, or time elements by selecting and completing the **Patch Criteria** text boxes. 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 don't wish to evaluate each file manually to determine if you want a patch to be built for it. Click the **Check All** button to mark the check boxes of all of the files in the list. Then, use the Patch Criteria to limit the building of patches to those files likely to produce a benefit.

Table 5.7 ~ 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. The default is a cleared (empty) check box.
Min Bytes Saved	Specify the minimum number of bytes that the patch must save. Otherwise, the patch will not be built. The default is a cleared (empty) check box.
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. The default is a cleared (empty) check box.

Once a component file has been selected, the buttons used for patch building are enabled (see Figure 5.32 below).

To apply a criterion, select its check box. The adjacent data entry fields become 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 qualify according to all of the criteria you choose to apply.

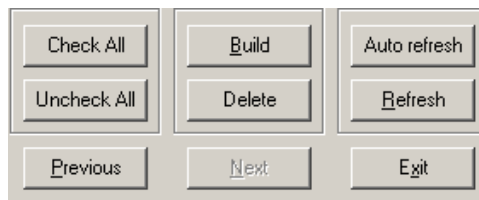


Figure 5.32 ~ Enabled buttons for patch building.

Table 5.8 below explains the button functions.

Table 5.8 ~ Button Functions

Button	Action
Check All	Selects any, and all, files listed in the Component File Name column.
Build	Click here to begin building patches.
Auto Refresh	Selecting Auto refresh opens the Auto Refresh Timer dialog box (see Figure 5.33 on page 160). When using Auto Refresh, enter a refresh interval between 1 and 99 minutes. You can cancel the auto refresh, by entering 0.
Uncheck All	De-selects any and all selected files in the Component File Name column.
Delete	Deletes a selected component file.

Table 5.8 ~ Button Functions

Button	Action
Refresh	Refreshes build status.
Previous	Returns you to the previous screen.
Next	When enabled, click Next to proceed to the next screen.
Exit	Click Exit to exit the Optimization Wizard.

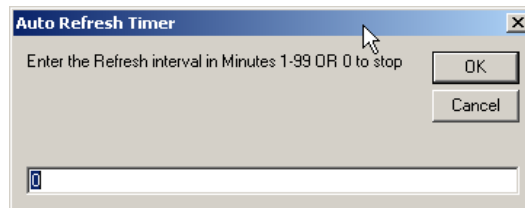


Figure 5.33 ~ Auto Refresh Timer dialog box—Enter an interval, in minutes, to automatically refresh the information in the Status column.

15. Select the check box next to the files you want to patch.

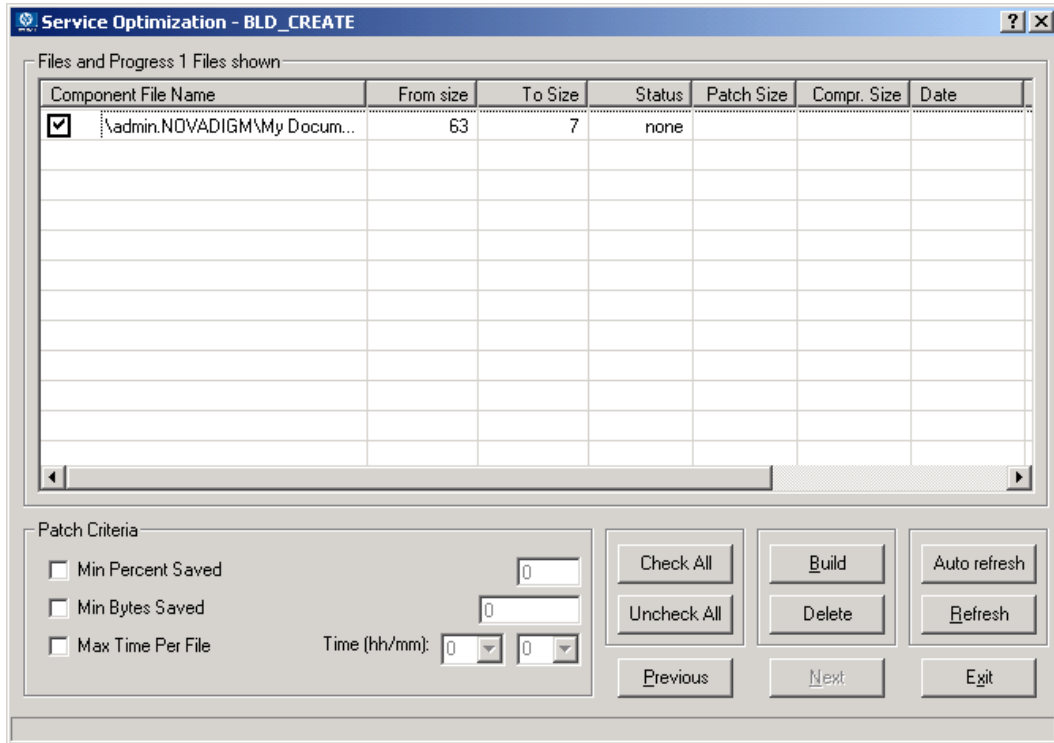


Figure 5.34 ~ Service Optimization—Selecting the files to build a patch.

- 16.** Click **Build**.
- 17.** The Radia System Explorer will prompt you to confirm the new patch build.

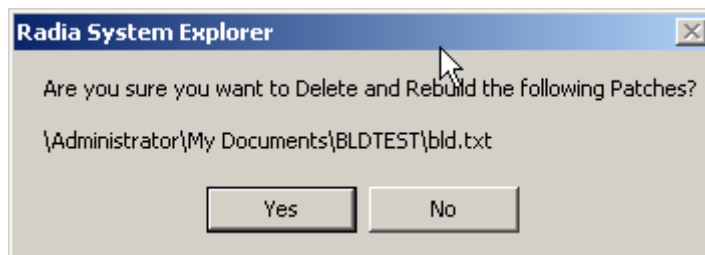


Figure 5.35 ~ Build confirmation dialog box.

18. 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 will then be sent to the Radia Configuration Server.

19. To update the **Status**, click **Refresh**.

20. Click **Exit** to exit the Service Optimization Wizard.

OR

Click **Previous** to return to the **Available Packages** dialog box to build additional patches.

The patch is located in the PATCH domain, under the PATCH class.

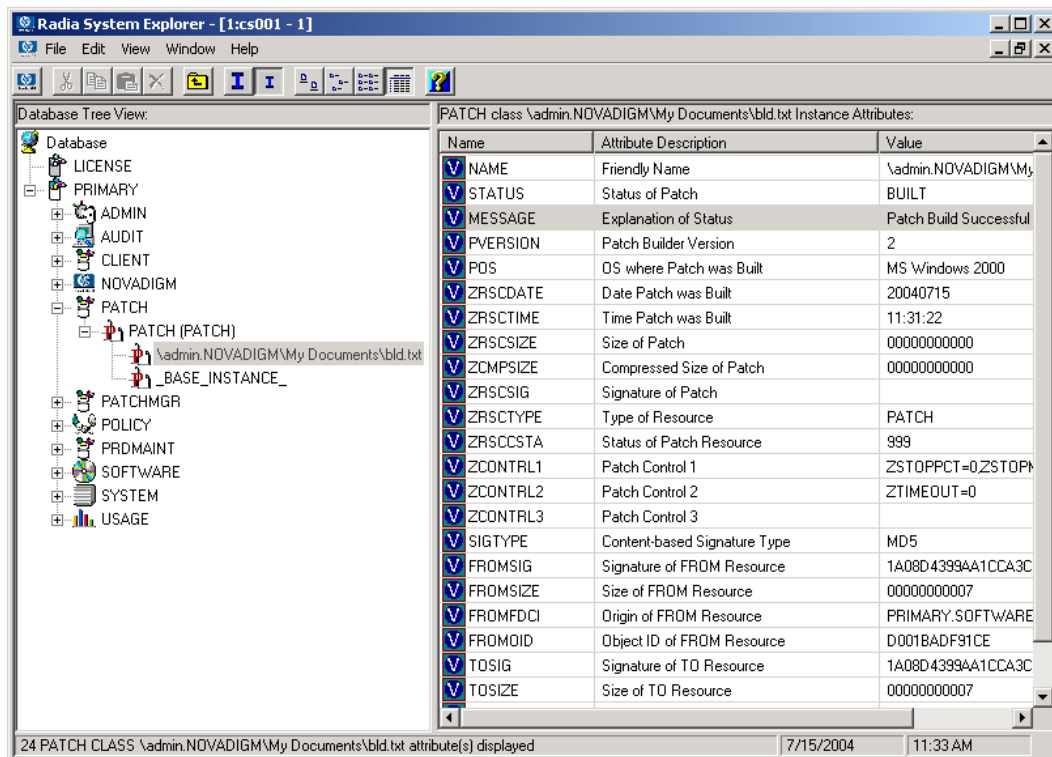


Figure 5.37 ~ The MESSAGE instance in the patch.

Radia Service Groups

Radia manages products that require more than one service package to establish full product installation or operation. You can use Radia Service Groups when a product requires other service-packages or has dependencies on other services. This includes MSI packaged products where:

- A product may utilize more than one MSI service-package, for example where MS Office may require other language pack services.
- A large product may need to be split into smaller sub-services to install only specific parts of the product suite.

Service grouping will allow multiple services to be grouped together by a higher, or **master**, service. This action allows administrators to logically group together similar or related components into one easy to understand collection.

For Radia version 3.x, use the Radia System Explorer to define and then group services and packages to form the Master service.

Note

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

- **The Radia Software Manager**
Each service within the Radia Service List will display only the Master service.
- **The Radia Application Manager**
The radskman command contains the logic to process multiple service requests. radskman will perform the request on each of the individual sub-services and report the final results back to the Master service.

Prerequisites

Prior to configuring a Master group, you need to change or confirm the following settings:

- **The Radia Configuration Server (version 4.5 or greater)**

In the EDMPROF.dat file verify that the

- ALLOW_DUPLICATE_INSTANCES in the MGR_OBJECT_RESOLUTION section is set to **No**.

■ The Radia Configuration Server (versions less than 4.5)

In the EDMPROF.dat file verify that the

- ALLOW_CIRCULAR_REFERENCE in the MGR_OBJECT_RESOLUTION section is set to **Yes**.

And,

- ALLOW_DUPLICATE_INSTANCES in the MGR_OBJECT_RESOLUTION section is set to **No**.

■ The Radia Configuration Server Database

- If you would like to configure a set of services (a grouping in which services depend on one another for installation), you will need to 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* beginning on page 61.

Table 5.9 ~ ZSVCGRP Variable

Value	Behavior
Y	Indicate Y (Yes) for the Radia Configuration Server to process the service groups as if they are unrelated applications or applications that are not dependent upon one another for installation.
D	Indicate D (dependent) for the Radia Configuration Server to process the services under the service group as though they are dependent upon one another for installation. If one sub-service fails, the remaining services will not be installed.
N	Indicate N (No) for no service groups.

If the variable is not present, the default processing will be as if the setting were set to Y.

Creating a Master Group

Once the changes have been made to the Radia Configuration Server EDMPROF.dat you can begin creating Master Groups.

To do this, you will use the Radia System Explorer to create a new Application (ZSERVICE) instance for the Master Group service. Next, for each service that is to be a member of the group, you will make a connection between the member service and the Master Group service instance. You can add service members to the master group service at any time.

For our example, we will be creating a master group called Sales Services. This master group will contain the Amortize and GS Calc services provided in the Radia sample database.

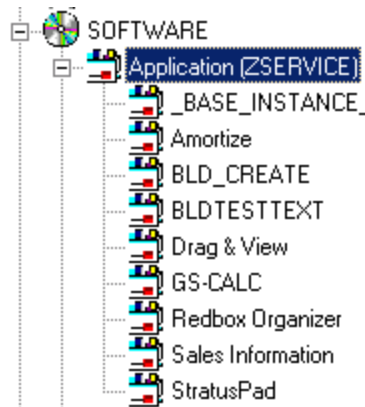


Figure 5.38 ~ ZSERVICE sample services.

To create a Master Group

1. If you have not already done so, access the Radia System Explorer and navigate to the **PRIMARY.SOFTWARE.ZSERVICE** class.
2. Expand the class to display the Application (ZSERVICE) instances.
3. Right-click on the Application (ZSERVICE). A shortcut menu opens.

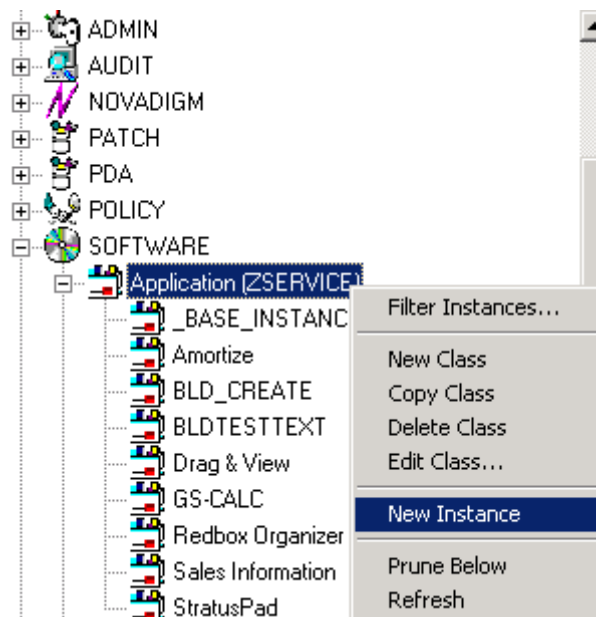


Figure 5.39 ~ Creating a new ZSERVICE instance.

4. Select **New Instance** from the shortcut menu. The **Create Instance** dialog box opens.
5. Enter the name of the new ZSERVICE instance you are creating; for our example, Sales Services.

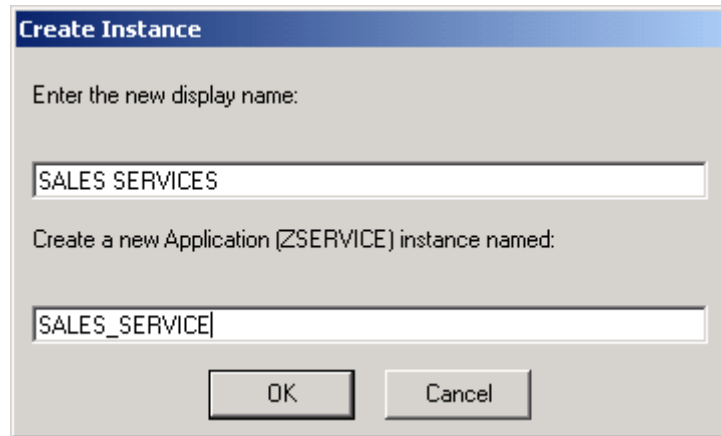


Figure 5.40 ~ Create a new instance – Sales Service.

6. Click **OK** to create the instance. The new instance appears in the Application (ZSERVICE) Class.

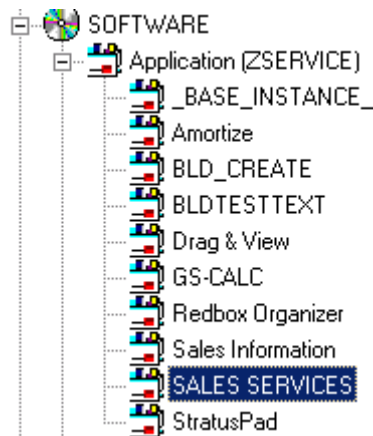


Figure 5.41 ~ Newly created instance—Sales Services.

You are now ready to begin adding services that will comprise your Master Group.

7. Right-click on SALES SERVICES, and select **Show Connections** from the shortcut menu.

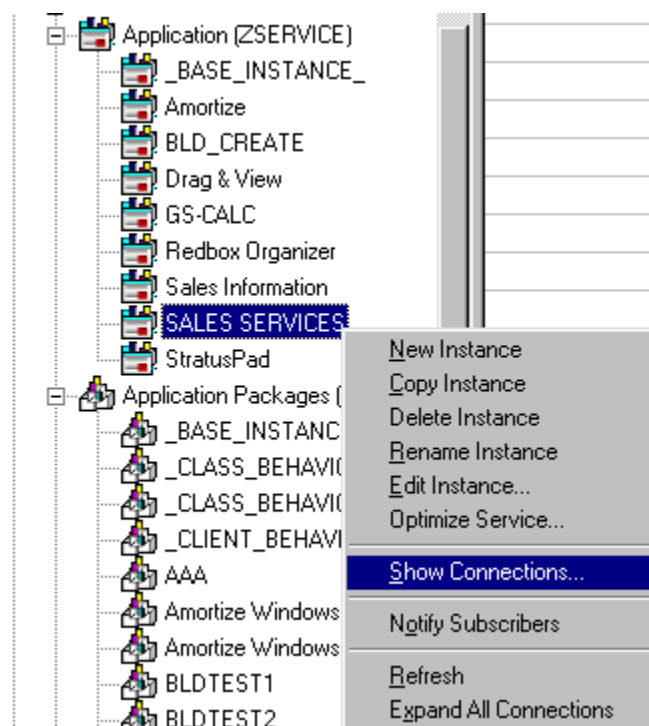


Figure 5.42 ~ Select Show Connections from the shortcut menu.

The **SOFTWARE.ZSERVICE Connections** dialog box opens.

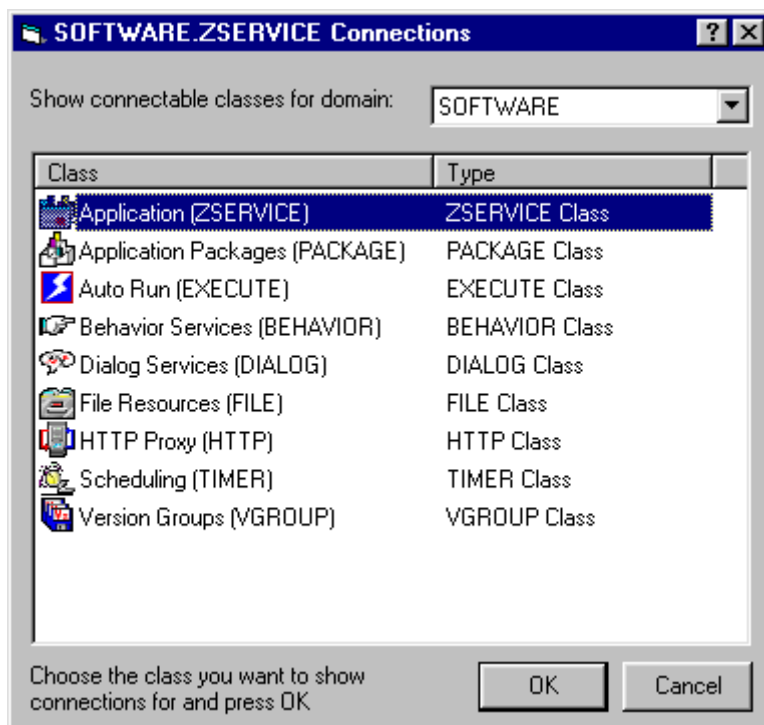


Figure 5.43 ~ *SOFTWARE.ZSERVICE Connections dialog box.*

8. Select **Application (ZSERVICE)** class from the **SOFTWARE.ZSERVICE connections** dialog box, and click **OK**.

A list of the **ZSERVICE** class instances is displayed on the list view of the screen.

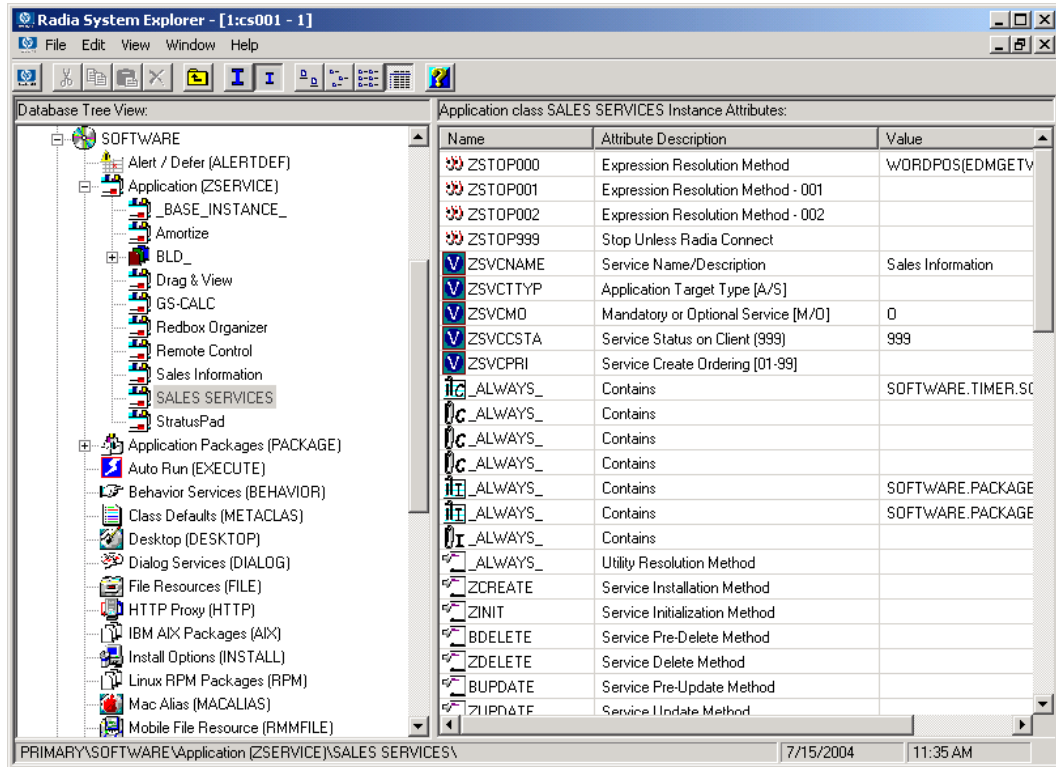


Figure 5.44 ~ The Application (ZSERVICE) Class instances are displayed in the list view.

9. Select the services from the right, and drag them to your SALES SERVICES ZSERVICE instance. Drop them onto the service when the cursor turns to a paper clip.
10. Confirm the connection.
11. Continue adding services, following the above outlined steps.

Your Master Group has been completed.

Note

The services in a Service Group will install in "visual order". This means, from the top downward as connected to the Master Service in the database. Services in a service group do not honor the settings in ZSVCPRI.

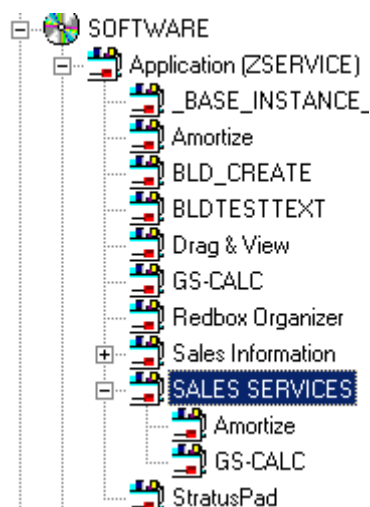


Figure 5.45 ~ The completed Master Group—Sales Services.

This Master Group service is now ready to be deployed to your users.

Radia Notify

Radia Notify enables the Radia administrator to cause one or more client computers to connect to the Radia Configuration Server and to update or remove a specified application. It can also send e-mail notification to client computers, based upon the applications Radia manages for them. See the *Radia Application Manager Guide*, the *Deploying Applications* chapter for Radia Notify requirements.

Notify communicates with client computers that are members of an audience list. A client computer is added to the audience list when Radia installs an application to that computer.

There are a number of prerequisites for using Notify:

- The subscriber must be connected to the Radia Configuration Server prior to the notification. Notify is designed to notify only clients whose information is in the PROFILE file in the Radia Database.
- The subscriber must have already installed the application using Radia in order to appear in the audience list for the specified application.
- If choosing to use e-mail to notify subscribers, then be sure that the e-mail address for the subscriber is in the EMAIL variable of the USER instance in the POLICY domain, USER class.
- The Radia Configuration Server must be properly configured for Notify. For information on setting the appropriate values for the Radia Configuration Server Settings file, EDMPROF.DAT, see the *TECHNOTE: Configuring the Radia Configuration Server for E-mail Notification* (Document ID: OV-ENKB01129) on the HP OpenView support web site.
- Select how the System Explorer should generate the Notify command for clients from the Notify Tab of the View Options dialog box. You can customize the UID and STARTDIR values when notifying subscribers to install an application. For details, refer to *Notify Tab* on page 40.

The Radia System Explorer offers two ways to initiate a Notify action:

- **Drag a Policy (USER, DEPT, and WORKGRP) class instance to an Application (ZSERVICE) instance.**
The audience list will consist of all client computers that are members of the Policy class instance and for which Radia 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 client computers for which Radia has installed and manages the application represented by the Application instance.

Note

In either case, Radia must have already installed the application on the client computer before that computer is eligible to be notified.

Once the Notify is initiated, the Radia System Explorer presents the same dialog boxes to complete the notification process, regardless of how the Notify was initiated.

To create a notification event

1. If you have not already done so, access the Radia System Explorer and navigate to the **PRIMARY.SOFTWARE.ZSERVICE** class.
2. Expand the Application (ZSERVICE) class.
3. Navigate to and expand the PRIMARY.POLICY class.
4. Initiate a notify action using a drag-and-drop operation or a menu-based selection.
 - **Drag-and-drop Notify**
To initiate a drag-and-drop Notify, select a Policy instance (workgroup, department, or individual user), drag it onto an Application instance, and drop it there.
 - **Menu-based Notify**
To initiate a menu-based Notify, right-click on an Application instance. The following menu opens.

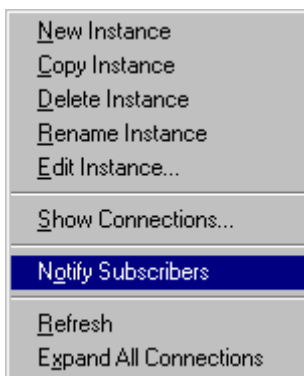


Figure 5.46 ~ Menu based notify—shortcut menu.

5. Select **Notify Subscribers**.

Notify Dialog Boxes

Regardless of how the Notify was initiated (drag-and-drop or menu-based) the following process takes place.

The Notify action takes the set of all subscribers to an application from the PROFILE file as the starting point for creating a notify audience. The following message illustrates this.

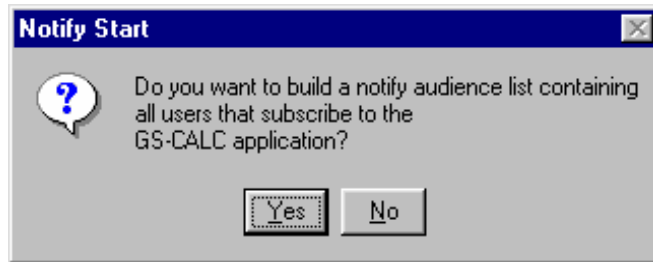


Figure 5.47 ~ Notify Start dialog box.

6. Select **Yes** to continue.

OR

Select **No** to cancel the Notify.

The Notify action retrieves the set of the subscribers from the POLICY domain. If the selected application does not have any subscribers, the following message opens.

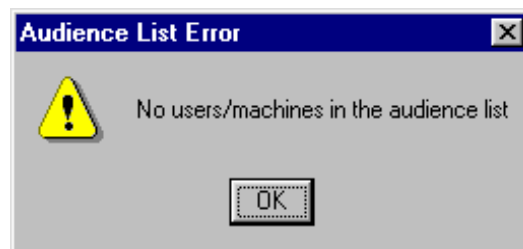


Figure 5.48 ~ Audience List Error.

The **Radia Notify Manager** dialog box opens.

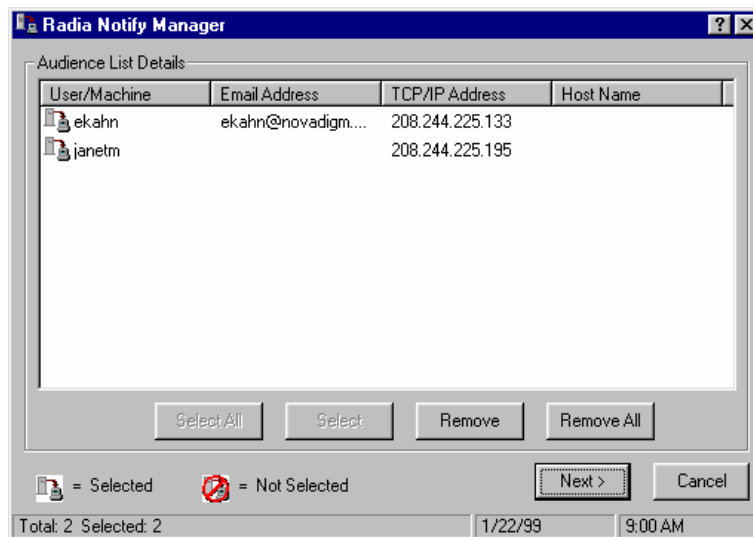


Figure 5.49 ~ Radia Notify Manager dialog box.

7. To notify the entire audience set, click **Select All**. This is the default when the audience list is first displayed.

To generate a tailored Notify audience, highlight the row associated with a subscriber and then choose either **Select** or **Remove**. Before using the **Select** button, you can click **Remove All** to clear the "selected" status of all client computers in the audience list.

The icons to the left of the subscriber name and in the lower left corner of the dialog box denote which subscribers have been selected. In addition, you can see the total number of client computers in the audience list, and the number currently selected, displayed at the very bottom left of the dialog box.

8. When you've finished selecting client computers in the Notify audience list, click **Next** to continue.

You can now select the type of Notify action you want. There are three options as shown in the following dialog box.

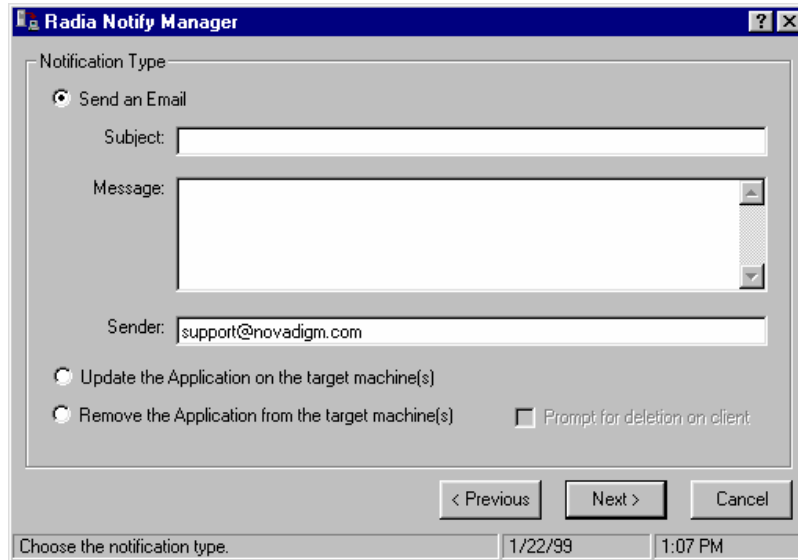


Figure 5.50 ~ Radia Notify Manager allows three options for notification.

Note

The @ symbol is required as part of the sender's address. An error will result if this symbol is not entered.

- **Send an Email**
Use this option to inform subscribers of an application's status or non-mandatory availability. The subscriber's e-mail attribute in the user instance must contain a valid entry.
- **Update the Application on the target machine(s)**
Use this option to install updates or new versions of an application. The updates or new versions must reside in the Radia Database.
- **Remove the Application from the target machine(s)**
Use this option to remove an application from the specified desktops.

The action you choose will be taken for all selected members of the audience list.

Send an Email Using Notify

If you choose **Send an Email**, and fill in the text boxes, the dialog box will resemble the following:

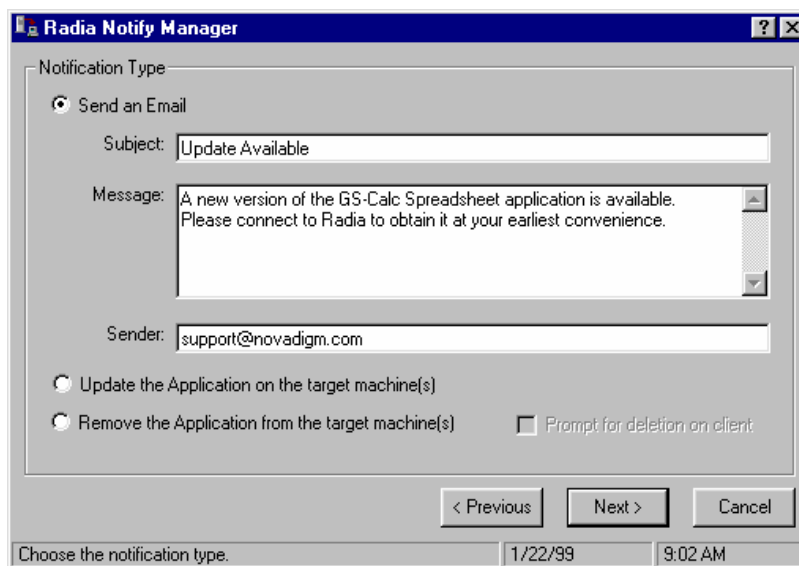


Figure 5.51 ~ Selecting the **Send an Email** option will display a view similar to this dialog box.

Update an Application Using Notify

To update the application on selected client computers in the audience list, choose **Update the Application on the target machine(s)**.

Remove an Application Using Notify

To remove the application from the selected client computers in the audience list, choose **Remove the Application from the target machine(s)**.

When you choose this option, the **Prompt for deletion on client** check box becomes enabled. Normally, Notify will remove the application from the selected client computers in the audience list without requesting permission from the end user. This allows unattended application removal from client computers.

To require the end user to give permission, select this check box. A dialog box will open on the client computer requesting permission to remove the application. If the end user agrees, Radia removes the application. If the end user denies permission, the application is left intact on the client computer.

- When you've finished composing the e-mail or selecting the notification type, click **Next** to continue.

The **Notification Details** dialog box opens.

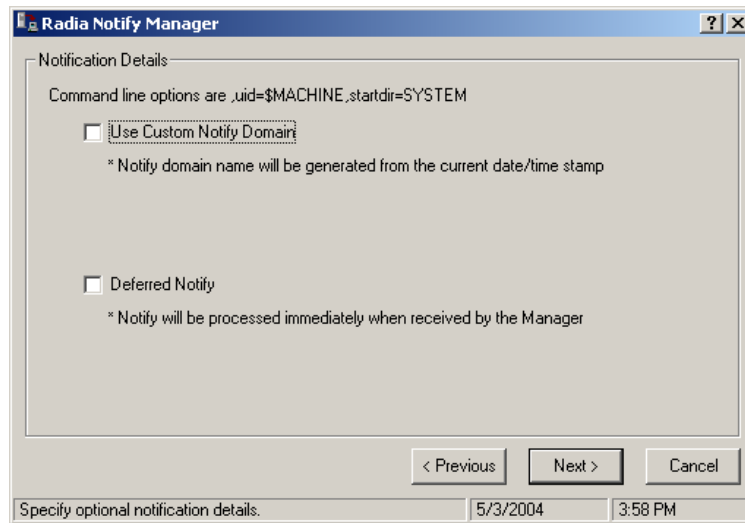


Figure 5.52 ~ After selecting the notification type, a Notification Details dialog box opens.

Set Notification Details

The top line on the **Notification Details** dialog box identifies the current Notify command line options for UID and STARTDIR that will be used. The Notify tab of the System Explorer Options dialog box allows these values to be customized.

Note

To reset the Notify command values for UID or STARTDIR values, you need to cancel out of this Radia Notify Subscriber dialog box, set the Notify Options for UID and STARTDIR using the View Options menu, and then repeat the Notify Subscribers selections. See *Notify Tab* on page 40 for more information.

The **Notification Details** dialog box allows you to specify a name for the domain in the NOTIFY file that will contain the results of the current notify operation. It also lets you specify when the notify operation will take place.

You may choose to leave the Notification Details check boxes empty. This will cause the default settings to be used. The defaults settings are:

- Notify domain name – will be generated from the current date/time stamp.
- Deferred notify date/time – will be processed immediately when received by the Radia Configuration Server.

To specify a domain name

1. Select the **Use Custom Notify Domain** check box. A text box opens.
2. Type a domain name.

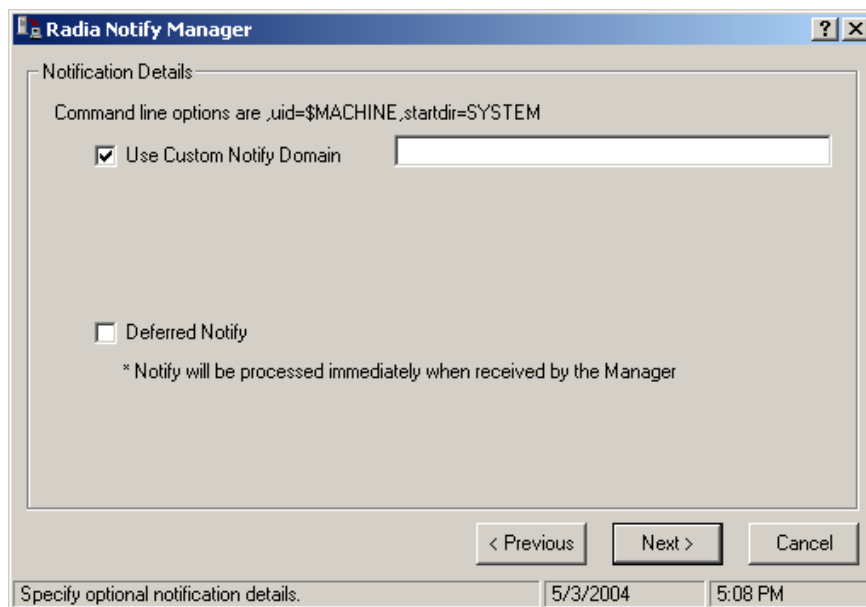


Figure 5.53 ~ Check the Use Custom Notify Domain box to specify a domain name.

You might want to do this if you rarely use the Notify feature and you'd like your domain names in the NOTIFY file to suggest the purpose of the notify operation. See *Notify File Structure* on page 186 for additional information.

To defer notification until a specific date and time

1. Select the **Deferred Notify** check box in the **Notification Details** dialog box. Additional controls appear as shown in Figure 5.54 below.

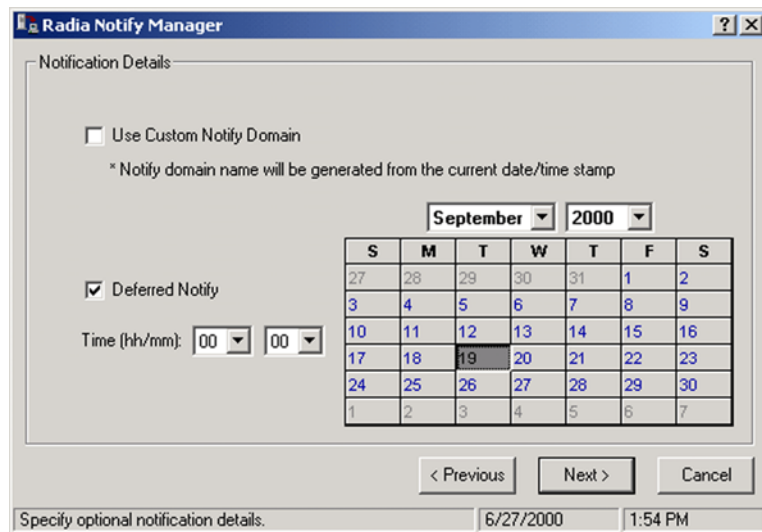


Figure 5.54 ~ Selecting the Deferred Notify box will display the time and date controls.

2. Enter the date and time when the notify operation should occur. The Radia Configuration Server must be running in order for the notify operation to happen. If not, the notify operation happens the next time the Radia Configuration Server starts, after the designated date and time.
3. Click **Next**.

A summary screen opens (see Figure 5.55 on page 183).

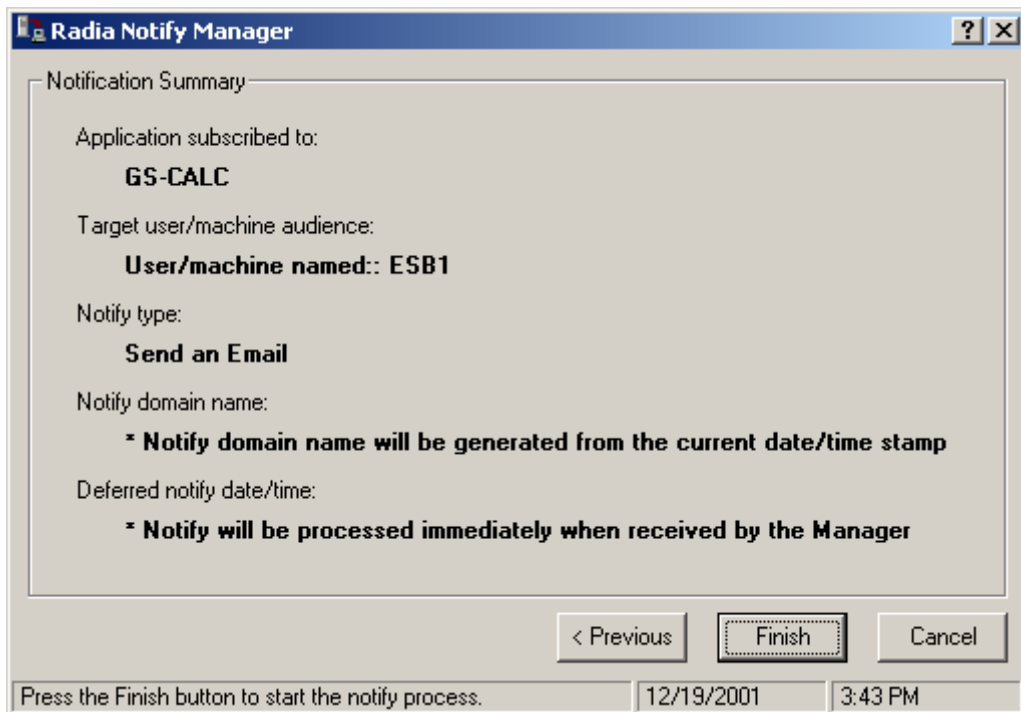


Figure 5.55 ~ After selecting the notification type and details, a summary screen opens.

4. Click **Finish** to start the Notify.

The Radia Configuration Server schedules the Notify and generates an object in the NOTIFY file using the date and time of the Notify action. The format of the object name is YYYY_MM_DD_HH_MM_SS as displayed in the following dialog box.

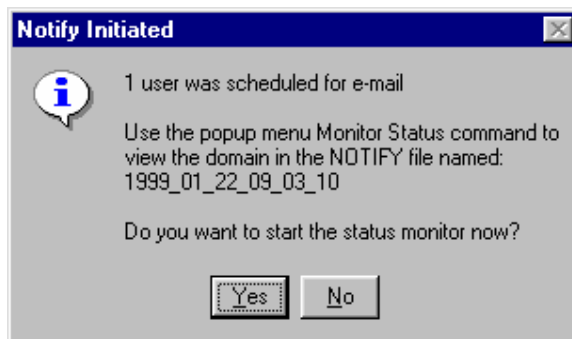


Figure 5.56 ~ Notify Initiated dialog box.

5. To view the status of the Notify, click **Yes**.

A **Notify Status** window opens.

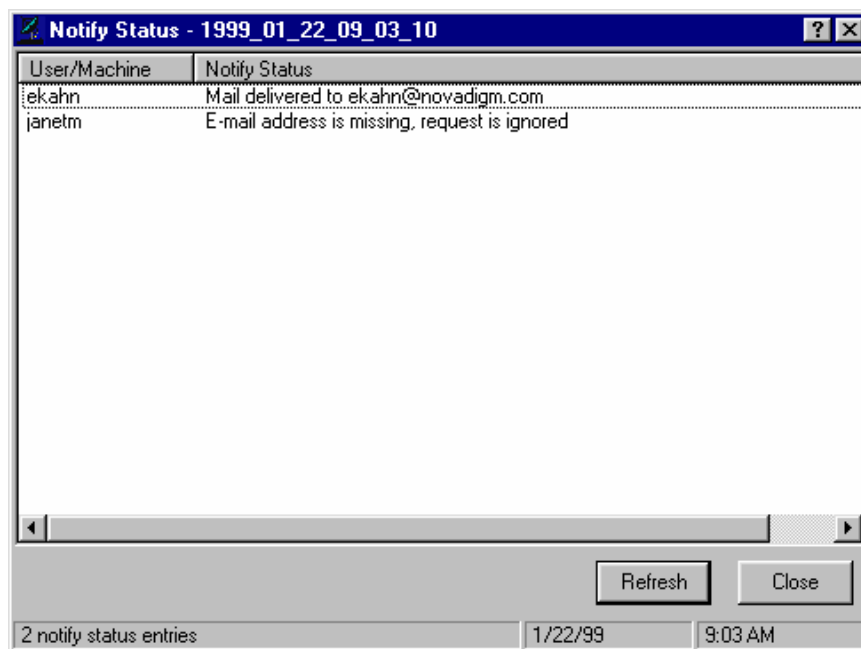


Figure 5.57 ~ The Notify Status dialog box displays the status of the notification.

The Status Monitor provides a snapshot of the progress of the Notify process.

6. Select **Refresh** to update the Status Monitor. It may take some time for all selected members of the audience list to be notified successfully or unsuccessfully. Refresh the Status Monitor to see the latest status information.

The information provided can be used to correct, reconfigure, and regenerate the Notify request so that all recipients receive the notification. Notify information is also written to the Radia Configuration Server log and can be viewed there.

Once a Notify has been initiated for the first time, the NOTIFY file will thereafter appear among the files of the Radia Database in the Radia System Explorer tree view.

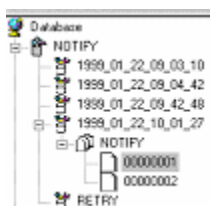


Figure 5.58 ~ After an initial notification, the Notify file is listed among the files of the Database in the Radia System Explorer tree view.

You can use Radia System Explorer to examine the domains, classes, and instances of the Notify file.

7. Right-click a domain in the Notify file.

The following shortcut menu opens.

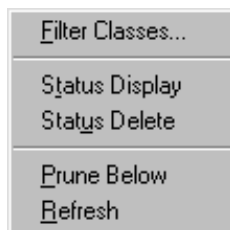


Figure 5.59 ~ Notify file—shortcut menu.

- Click **Status Display** to view the status of the associated Notify operation.

- Click **Status Delete** to remove the status information associated with the Notify operation from the Radia Database.

Notify File Structure

The Notify file is divided into domains, where each domain is one Notify operation. The name of the domain is in the form

YYYY_MM_DD_HH_MM_SS

which represents the date and time when the Notify operation was initiated. The opened NOTIFY domain in Figure 5.58 on page 185,

1999_01_22_10_01_27

holds the results of a Notify operation that was initiated on January 22, 1999 at 10:01:27.

Each Notify file domain has one NOTIFY class. Each NOTIFY class contains an instance for each selected member of that Notify operation's audience list. The instances are named with eight-digit numbers starting with 00000001 and running sequentially up to the total number of selected members in that Notify operation's audience list.

Each instance contains attributes that identify the selected audience list member, the kind of Notify operation (see the req= parameter of the command line in the NTFYCMDL attribute), and the results of the Notify operation with respect to that member (see the NTFYMSG and NTFYRC attributes).

Retrying a Notify Operation

Often, not every selected audience list member can be notified successfully when the Notify operation is initiated. At that time, one or more selected client computers may be turned off, may not be running the Radia Notify module, or may not be accessible via the normal communication channel, or may be broken, etc. An unsuccessful Notify attempt creates an instance in the RETRY domain of the Notify file.

Radia can automatically retry the Notify operation for failed Notify attempts. To do so, the Radia Configuration Server must be started with the Notify Retry Manager (zrtrymgr module), as indicated in the following excerpt from the Radia Configuration Server NVDPROF.DAT file.

```

[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexmgr) RESTART=YES
CMD_LINE=(zsnmpmgr) RESTART=YES
CMD_LINE=(zsmtrmgr) RESTART=YES
CMD_LINE=(zsmtsmgr) RESTART=YES
CMD_LINE=(zntytmgr) RESTART=YES
CMD_LINE=(ztcpmgr) RESTART=YES
CMD_LINE=(zrtrymgr) RESTART=YES

```

Include this line to enable Notify retries.

Figure 5.60 ~ Excerpt from the Radia Configuration Server EDMPROF.DAT file.

The Notify Retry Manager periodically examines the Notify file's RETRY domain, and re-attempts the Notify operation for each instance it finds there. To see the status of retry attempts, use Radia System Explorer to examine the contents of the RETRY domain. Right-click on the RETRY domain, and click **Status Display** from the shortcut menu.

Drag-and-Drop Notify for Wake-On-LAN Clients

Radia can issue a **wake-up** packet that will remotely **power-on** client computers that have been configured for Wake-On-LAN (WOL). The ease and straight-forwardness of the drag-and-drop functionality dramatically simplifies assigning notify eligibility to WOL clients. Using the Radia System Explorer, it is easy to connect Notify instances to a USER, WRKGRP, DEPT, and SERVICE.

In order to take advantage of the drag-and-drop feature to assign clients for WOL Notify, use the Radia System Explorer. Once you have accessed the Radia System Explorer, you can simply click and hold a Notify instance, drag it to the instance to which you want it connected, and release (drop) it, thereby creating the connection.

Refer to the *Radia Application Manager Guide* for information on how to configure a WOL client for drag-and-drop notify. Refer to the *Radia Configuration Server Guide* for information on changes to the Radia Configuration Server Settings file.

Filtering—Viewing Portions of the Radia Database

As your distribution model grows, you will find it convenient to restrict the Radia System Explorer to view only a portion of the Radia Database. This saves effort in scrolling through the window to locate the domain, class, or instance you need to inspect or edit. The Radia System Explorer offers the ability to set filters to provide user-defined restricted views of the Radia

Database. 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 Radia System Explorer 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 5.10 ~ 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 two Ls.
-??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 Radia System Explorer session (*permanent filters*), or filters that apply only to the current Radia System Explorer window (*temporary filters*).

Permanent Filters

Permanent filters are applied every time you open any Radia System Explorer window during the current Radia System Explorer session. Each class can have its own permanent filter specification during an Radia System Explorer session.

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. The classes of the POLICY domain will be listed.

2. Scroll down the tree view until you reach the USER class.
3. Right-click the USER class.

The following shortcut menu opens.

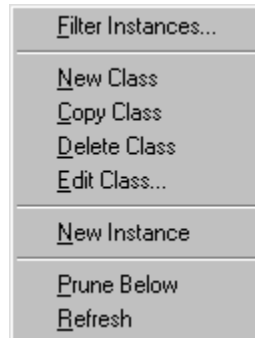


Figure 5.61 ~ Class shortcut menu.

4. Click **Filter Instances** to open the **Filter Instances** dialog box.

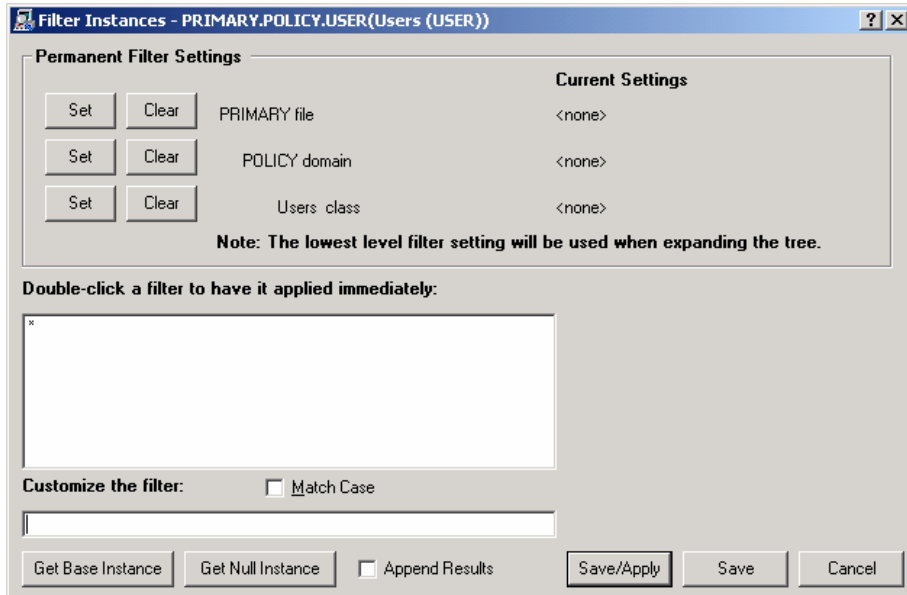


Figure 5.62 ~ Filter Instances dialog box.

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.

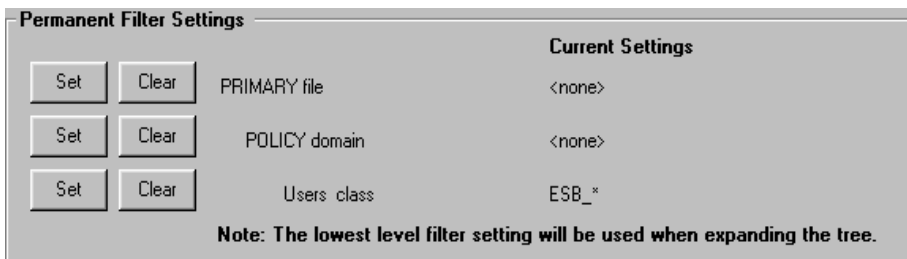


Figure 5.63 ~ Filter specification for the Users class is displayed under Current Settings.

7. To apply the filter, click **Save/Apply** at the bottom of the dialog box.

The dialog box will close, and the tree display will be updated with the filter applied.

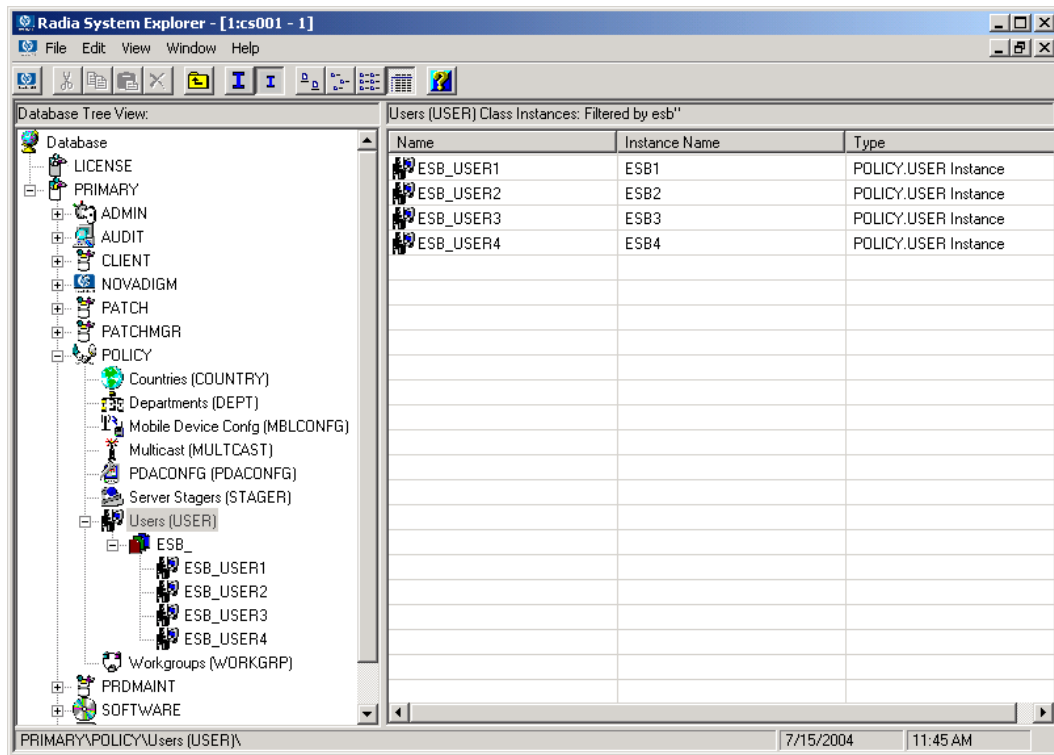


Figure 5.64 ~ Filter applied to the Users class.

This filter will be applied to the USER class in any additional Radia System Explorer windows you open during the current Radia System Explorer session.

To remove a permanent filter

1. Click the appropriate **Clear** button in the **Filter Instances** dialog box.
2. Click **Save/Apply** to remove the filter.

Temporary Filters

The **Filter Instances** dialog box also allows you to apply *temporary filters*. Temporary filters remain in effect only as long as the current Radia System Explorer window is open. You can accomplish this by entering a filter specification in the **Customize the filter field**, and then clicking **Save/Apply**.

To set a temporary filter to filter instances of the USER class on the prefix ESB_

1. Expand the POLICY domain in the tree view. The POLICY domain classes are listed.
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. Type ESB_* in the **Customize the filter** text box.
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. The POLICY domain classes are listed.
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 Instance** to view the `_BASE_INSTANCE_` of the USER class or click **Get Null Instance** to view the `_NULL_INSTANCE_` of the USER class.
6. Click **Save/Apply**.

Appending Filter Results

You may want to maintain viewing the current filter's result, and add an additional filter as well. To view both results at the same time, enable the **Append Results** checkbox as shown in *Figure 5.62 ~ Filter Instances dialog box* on page 190 when you perform the next filter.

Filtering Component Class Instances

Component class instances (such as those of the FILE, DESKTOP, REGISTRY, etc., classes) can be filtered based upon 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 presents a scrolling list labeled **Owning Package:**, as shown in *Figure 5.65* below. The package instances are listed in this scrolling list.

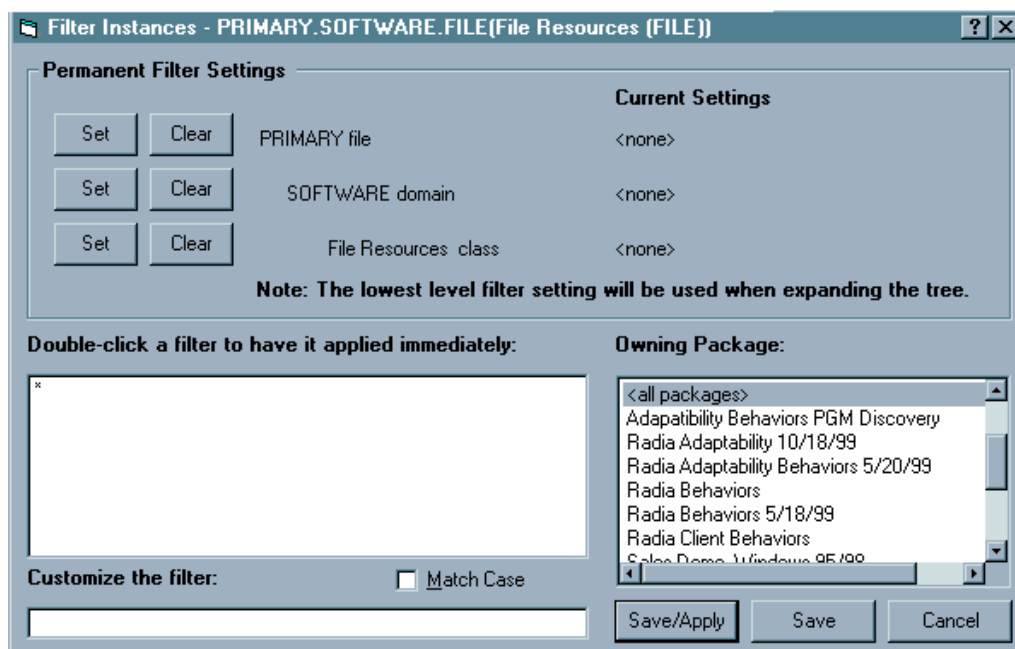


Figure 5.65 ~ Filtering component classes like the File class produces this dialog box.

To filter FILE instances to select only those that are connected to a particular PACKAGE instance

- Double-click the owning package in the **Owning Package** list.

OR

- Click the owning package in the **Owning Package** list, and click **Save/Apply**.

For example, select **Amortize Windows 95/98** 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 Radia System Explorer window.










File Resources (FILE) Class Instances:		
Name	Instance Name	Type
 Amortize Windows 95/98:\Amortize	DABC407A05F8_ADBBF78B	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_31607370	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_163AB498	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_7ACBD39D	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_46349AC5	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_A55546ED	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_7D86B40A	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_7B4E824A	SOFTWARE.FILE Instance
 Amortize Windows 95/98:\Amortiz...	DABC407A05F8_17BFE54F	SOFTWARE.FILE Instance

Figure 5.66 ~ This figure shows all of the FILE class instances that were filtered using the Owning Package list.

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 a Radia system 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 currently open Radia System Explorer window.

Customizing Express Filters

You can specify the lists of express filters that appear for selection in the **Filter Instances** dialog box by storing them in the Radia Database, and making any needed connections. The Radia System Explorer retrieves the list of express filters for a particular **Filter Instances** dialog box from the Radia Database as the dialog box opens.

The Radia System Explorer locates the correct list in the Radia Database by executing a resolution process based upon the user ID entered when the Radia System Explorer session was started, and the Radia component (i.e., domain, class, or instance) being filtered. Refer to *Understanding the Resolution Process* starting on page 199.

The list of express filters for a particular domain is drawn from an instance of the ZLIST32 class of the ADMIN domain in the PRIMARY file. Here's how.

Assume you've 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.

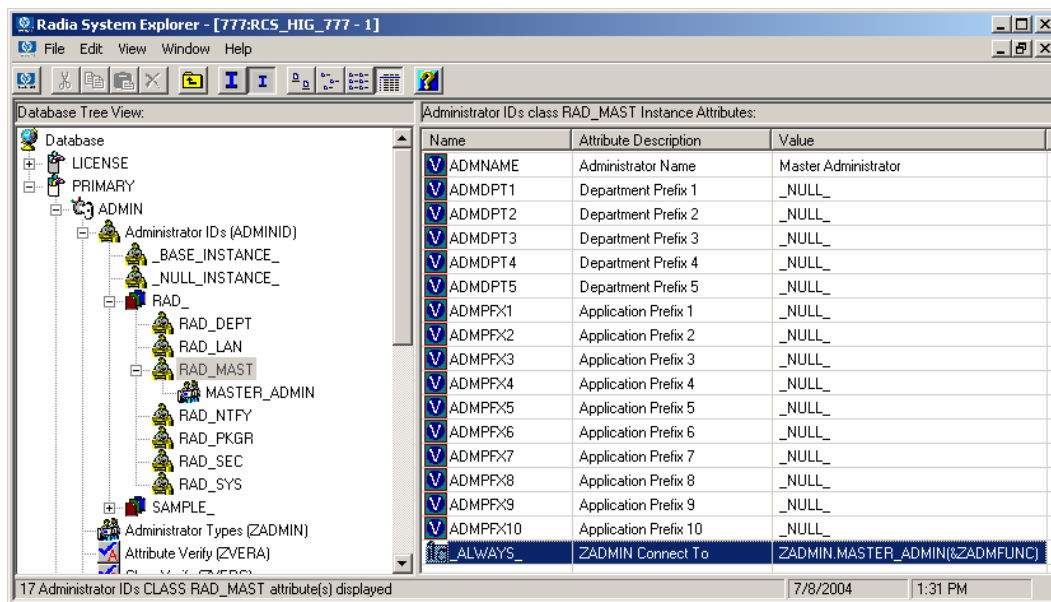


Figure 5.67 ~ The RAD_MAST instance has a connection to an instance of the ZADMIN class as displayed in the list view of the Radia System Explorer window.

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 execution. The Radia System Explorer sets the parameter's value before initiating the resolution. In this case, the Radia System Explorer 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.

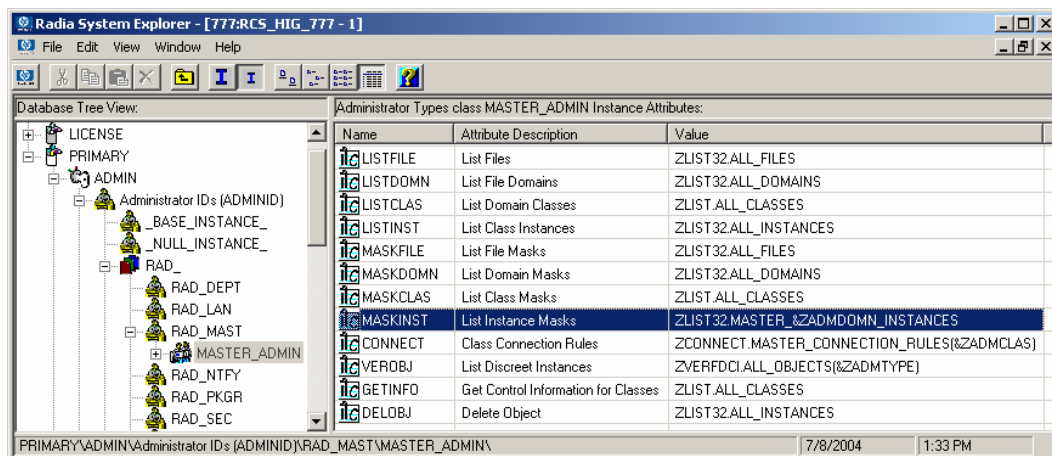


Figure 5.68 ~ The value of the ZADMFUNC variable is set to MASKINST to retrieve a list of instance masks.

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 Radia System Explorer, based upon which class the administrator right-clicked to access the **Filter Instances** dialog box. In the case of 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.

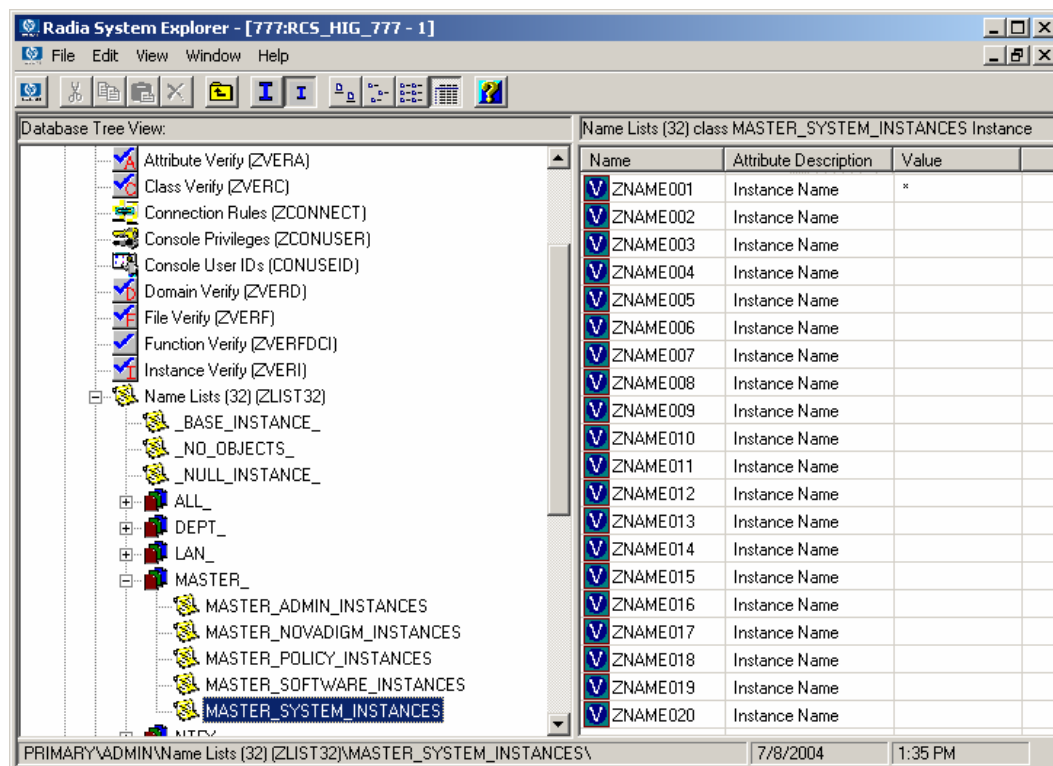


Figure 5.69 ~ A connection is further made to ZLIST32.MASTER_SYSTEM_INSTANCES.

To change the list of express filters for a particular domain, simply 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 73 and Editing an Instance on page 75 for information on how to use the Radia System Explorer to make these changes.

Summary

- The New Application Wizard automates the creation of an Application (ZSERVICE) class instance from a PACKAGE class instance.
- The Version Group Editor enables you to create, edit, and delete version instances associated with the Version Group. It controls the Version Group's deployment sequence.
- Service Optimization uses byte level differencing and its ability to generate patches to recreate original data.
- Radia manages products that require more than one service package to establish full product installation for operation. Create a Radia Service Group when a product requires other service packages or has dependencies on other services.
- Radia Notify enables the Radia administrator to cause one or more client computers to connect to the Radia Configuration Server and to update or remove a specified application.
- Radia can issue a wake-up packet that will remotely power on client computers that have been configured for Wake-On-LAN.
- The Radia System Explorer offers the ability of setting filters, and appending filter results, to provide user-defined restricted views of the Radia Database.

Understanding the Resolution Process

At the end of this chapter, you will:

- Understand the resolution process.

The Resolution Process

The Radia Configuration Server uses a procedure called the *resolution process* to accomplish a unit of work in response to a service request. The unit of work is defined by the contents of the Radia Database and parameters included in the request. In other words, what Radia does depends upon what information is stored in its Database, and what information accompanies the request for Radia to perform some action.

For example, the Radia Client Connect submits service requests to the Radia Configuration Server, and the Radia Configuration Server performs a *resolution process* in response to each such request.

Heaps and Objects

One way that the Radia Configuration Server recognizes a service request is when it receives an *object*. An object is simply a storage structure, or a container, for a record or a record set. An object consists of *instances* also known as *heaps*. An instance is an occurrence of a set of variables. Objects can consist of a single instance, such as the ZMASTER object or several instances, such as the FILE object.

You can inspect and even modify desktop objects using the Radia Client Explorer.

To access the Radia Client Explorer

- Click **Start, Programs**, and from the **Radia Administrator Workstation** menu choose **Radia Client Explorer**.

The **Radia Client Explorer** opens.

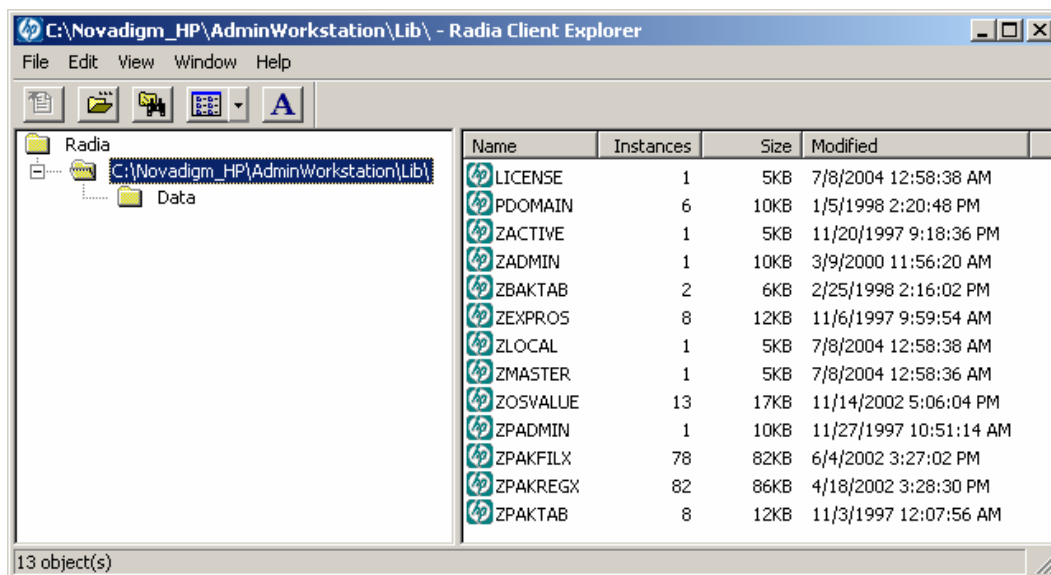


Figure 6.1 ~ The list view in the Radia Client Explorer displays the instance name, the number of heaps or instances, the size of the instance, and the last date and time the instance was modified.

Figure 6.1 above displays the Radia Client Explorer at its highest level. The Radia Client Explorer is used to view or edit objects. A user with the Radia Administrator Workstation installed can use the Radia Client Explorer as a diagnostic utility to view local objects on the desktop, edit objects, or create new objects.

To view an instance, double-click on that instance. The instance information will be displayed as shown in Figure 6.2 on page 202.

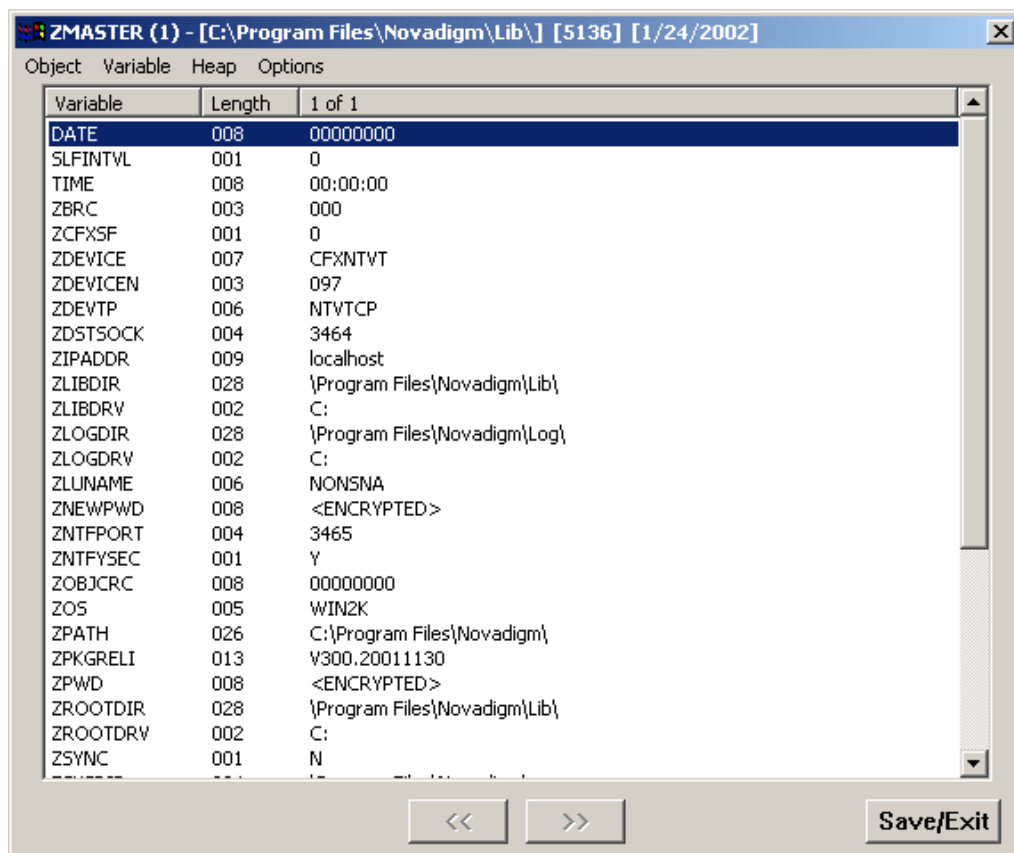


Figure 6.2 ~ Double-click on an instance to display the objects inside.

The resolution process uses the objects that are stored for each client to bring them to the *desired state*. The desired state embodies the content that Radia manages for a specific end-user computer. A model representing the desired state for each end-user computer is stored in the Radia Database. The desired state model is created and managed using the Radia System Explorer.

Resolution Process – Where Does It Begin?

As previously stated, the resolution process begins the moment a client computer connects to Radia. The ZMASTER is the first object sent to the Radia Configuration Server during the Client Connect. The ZMASTER object contains information about the client computer necessary to run Radia, such as the identity of the subscriber and the IP address of the client computer.

The Radia Configuration Server stores the ZMASTER object in a transient storage area called *global memory*. The resolution process defines variable attributes into objects contained in global memory, and their current values are maintained there. The current value of a variable attribute is the value it holds as of its most recent reference during the resolution process.

After storing ZMASTER in global memory, the Radia Configuration Server finds the Process instance for the ZMASTER. This is the *process entry point*. Its location is SYSTEM.PROCESS.ZMASTER.

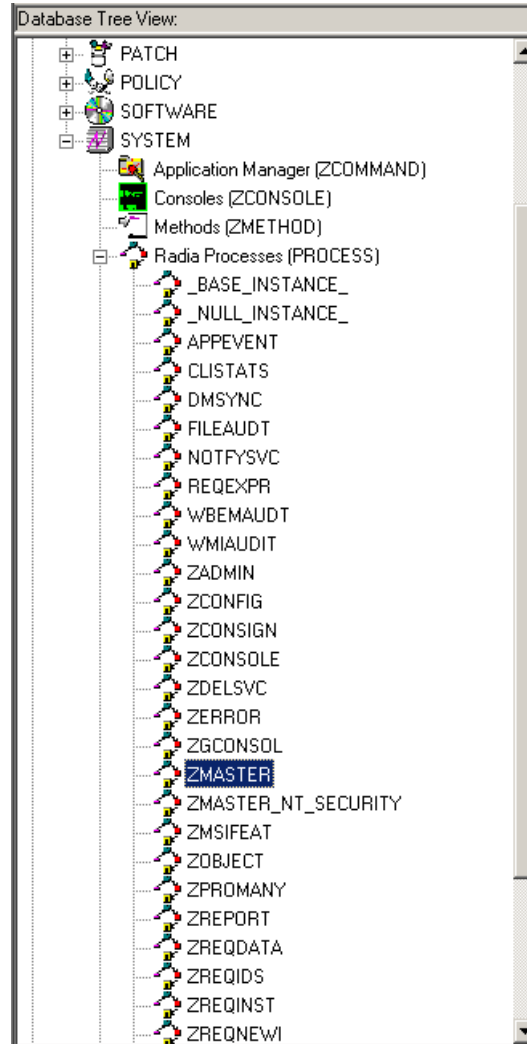


Figure 6.3 ~ Location of the ZMASTER object in the Radia Database.

The Radia Configuration Server reads each attribute of `SYSTEM.PROCESS.ZMASTER`. As a result of reading an attribute, the Radia Configuration Server may:

- Set variable values.
- Evaluate an expression.
- Execute a method.

- Connect to other instances.

If there is a connection to another instance, the Radia Configuration Server processes the connected instance. Then, the resolution process resumes with the referring instance at the next attribute after the connection attribute.

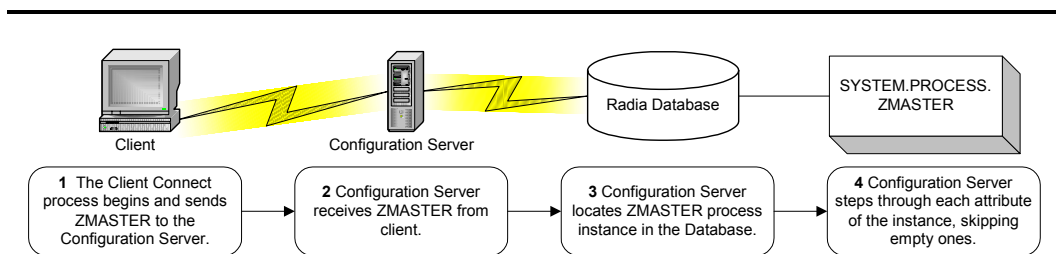


Figure 6.4 ~ The Radia Configuration Server performs resolution.

Resolution Example

Let's look at how a client's ZMASTER object is resolved.

In Figure 6.5 on page 206 the first connection instance links to `POLICY.USER.&(ZMASTER.ZUSERID)`. After processing `POLICY.USER.&(ZMASTER.ZUSERID)`, the resolution process will return to `PRIMARY.SYSTEM.PROCESS.ZMASTER`, and will process the next attribute which is a connection instance to `SYSTEM.ZMETHOD.PUTPROF_ZMASTER`.














Radia Processes class ZMASTER Instance Attributes:		
Name	Attribute Description	Value
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Connect To	
 _ALWAYS_	Connect To	POLICY.USER.&(ZMASTER.ZUSERID)
 _ALWAYS_	Method	SYSTEM.ZMETHOD.PUTPROF_ZMASTER
 _ALWAYS_	Method	SYSTEM.ZMETHOD.PUTPROF_ZCONFIG
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 _ALWAYS_	Method	
 DESCRIPT	Process Description	Processing Client Request for &ZCUIROBJ
 ZMAXDKRC	Max acceptable method Return Code	008

Figure 6.5 ~ The PRIMARY.SYSTEM.PROCESS.ZMASTER instance.

Resolution Process and Symbolic Substitution

During resolution, the Radia Configuration Server performs *symbolic substitution* to set values and connect to other instances. When processing symbolic substitutions, Radia will substitute the value of the second variable to replace the reference in the first variable.

References to be processed with symbolic substitution are specified using an initial ampersand.

For example, in SYSTEM.PROCESS.ZMASTER there is an always connection to POLICY.USER.&(ZMASTER.ZUSERID). The Radia Configuration Server substitutes the value of ZUSERID from the ZMASTER object that is in global memory. In Figure 6.6, the value of ZUSERID is ALEE. The resolution process will connect to POLICY.USER.ALEE to begin resolving the ALEE instance.

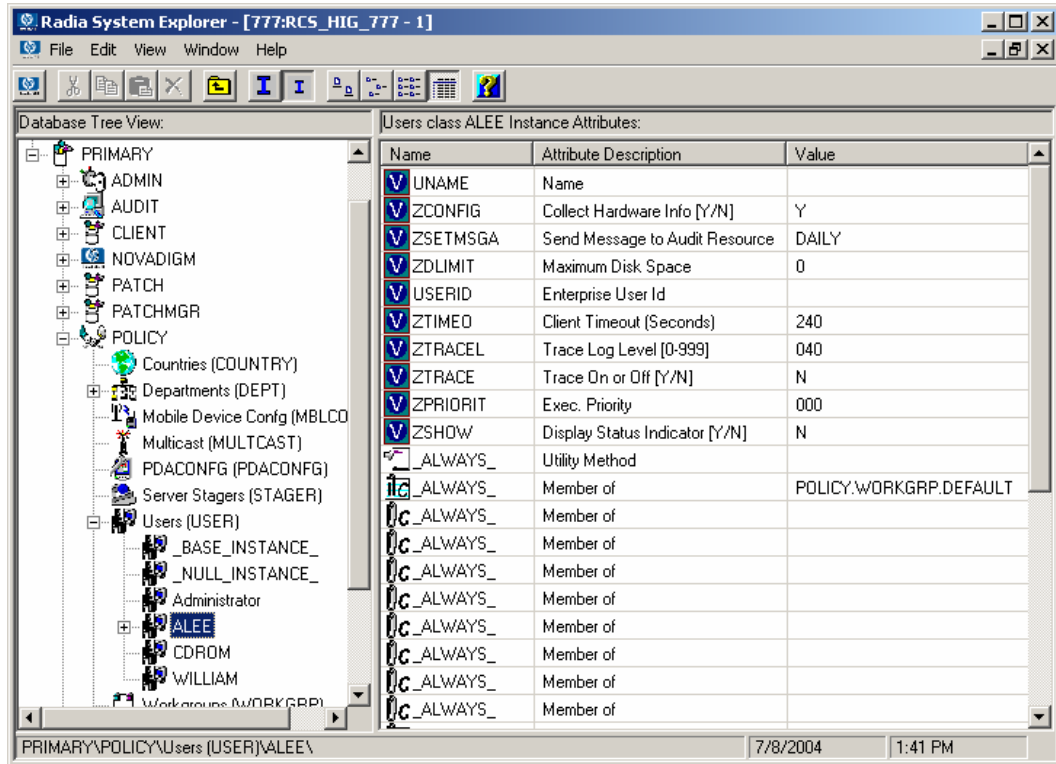


Figure 6.6 ~ POLICY.USER.ALEE instance.

In Figure 6.6, after setting a number of variables, the first connection attribute is to SOFTWARE.ZSERVICE.AMORTIZE. In your implementation, the POLICY instance may connect to a workgroup or a department that would then connect to a service.

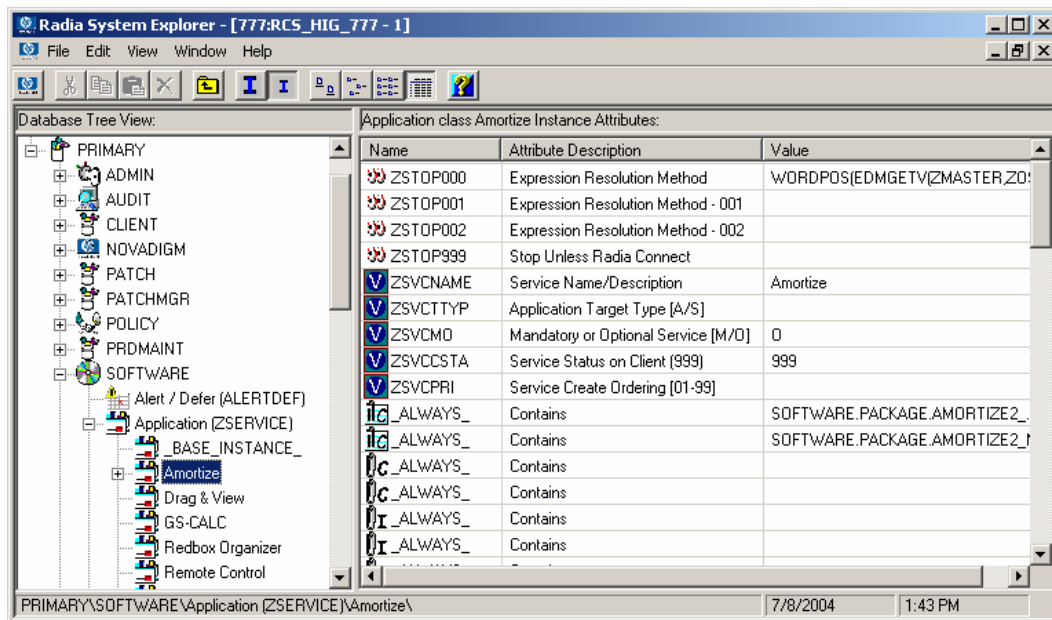


Figure 6.7 ~ The ZSERVICE.AMORTIZE instance.

The resolution proceeds to the service instance indicated in the first connection of the USER as shown in Figure 6.7 above. The Amortize service resolution begins with a ZSTOP expression variable. This expression variable contains statements that, if evaluated to "true", stop the resolution of the current instance. An expression allows alternative paths to be taken in a given resolution, based upon variable data. In this case, the expression checks to be sure that the operating system of the client computer is allowed for the Amortize software. If the client computer's operating system is either Windows 95 or Windows 98, the Resolution process continues with this instance, connecting the file instances, registry entries, path instances, and shortcuts. If the client has another operating system, resolution returns to SOFTWARE.ZSERVICE.AMORTIZE, and to the next connection instance.

Eventually, the resolution process will return to the User instance, finish resolving it and return to the process entry point, SYSTEM.ZPROCESS.ZMASTER. Referring back to Figure 6.6 on page 207, the next attribute connects to the PUTPROF_ZMASTER method. A *method* is a program that performs functions based on certain parameters.

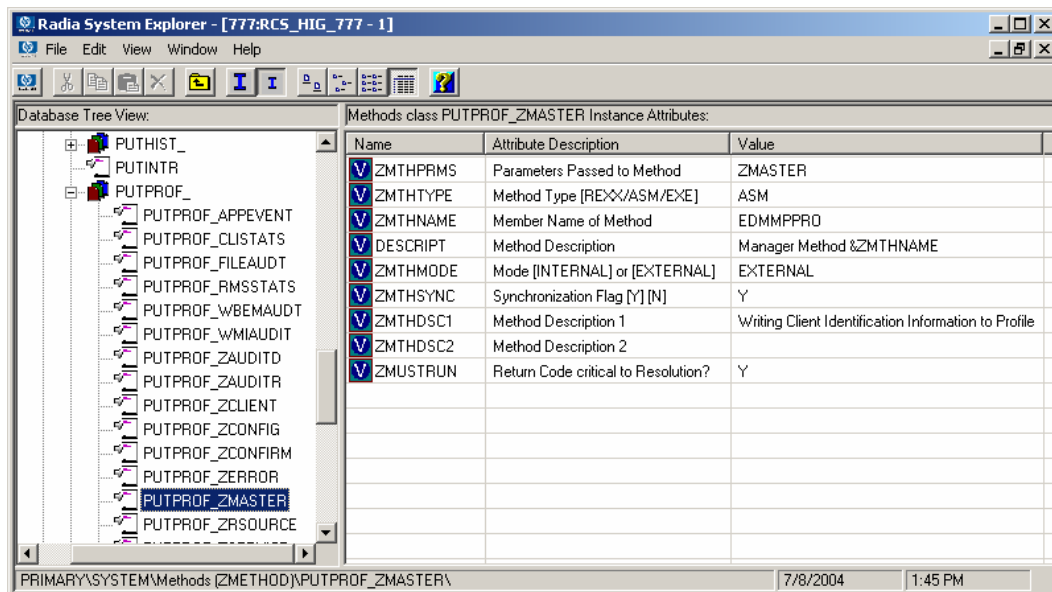


Figure 6.8 ~ The ZMETHOD.PUTPROF_ZMASTER instance.

The Radia Configuration Server executes the EDMMPPRO method, passing ZMASTER as a parameter. This causes the contents of the ZMASTER object in global memory to be written to the PROFILE file of the Radia Database.

After completing the processing of all attributes in the SYSTEM.PROCESS.ZMASTER instance, resolution terminates.

Summary

- The Radia Configuration Server uses a procedure called the *resolution process* to accomplish a unit of work in response to a service request.
- One way that the Radia Configuration Server recognizes a service request is when it receives an *object*.
- The Radia Client Explorer is one of two tools that can be used to view or edit objects.
- The resolution process uses the objects that are stored for each client to bring them to the *desired state*.
- The ZMASTER is the first object sent to the Radia Configuration Server during the Client Connect.
- During resolution, the Radia Configuration Server performs *symbolic substitution* to set values and connect to other instances.

Maintaining Connection Rules in the Radia Database

At the end of this chapter, you will:

- Understand connection rules.
- Know how to enable connection rules.

Introduction

The Radia System Explorer provides easy and reliable drag-and-drop connections between instances of classes. These connections, which contain connection rules, are stored in the Radia Database.

Connection Rules Overview

In order to allow for drag-and-drop capabilities for an added class, it is necessary to establish rules for the new class to follow. These connection rules are stored in the PRIMARY.ADMIN domain in the **MASTER_CONNECTION_RULES** instance of the **Connection Rules (ZCONNECT)** class.

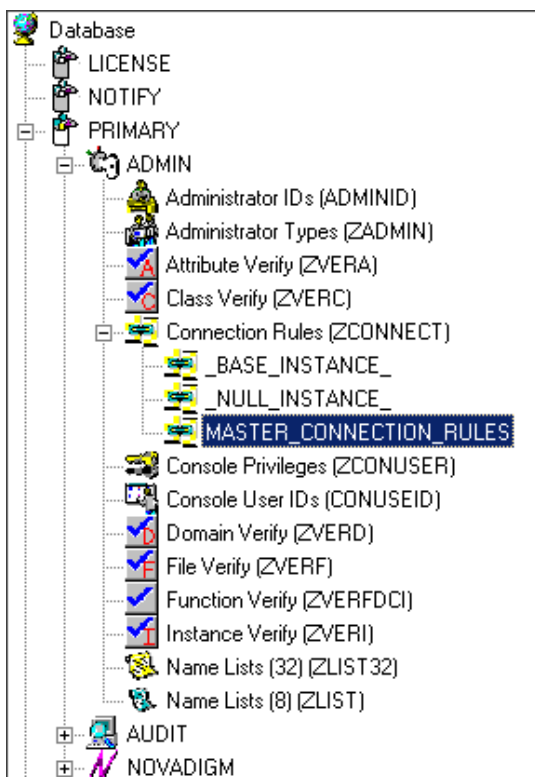


Figure 7.1 ~ The **MASTER_CONNECTION_RULES** instance of the **PRIMARY** file **ADMIN** domain **Connection Rules (ZCONNECT)** class is highlighted.

While the ZCONNECT class houses 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 will have to first create the rule in the ZLIST class. Then you will add the rule to the ZCONNECT class.

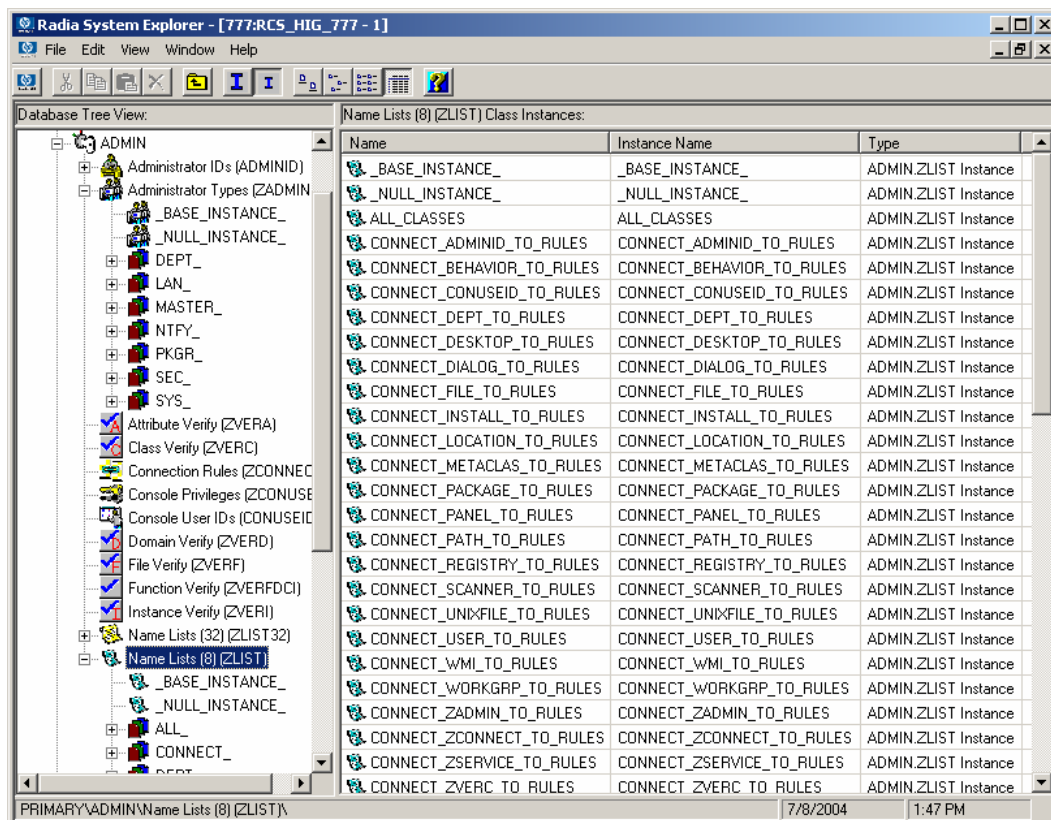


Figure 7.2 ~ The Name Lists (ZLIST) class in the ADMIN domain.

The Radia System Explorer 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 7.3 on page 214.

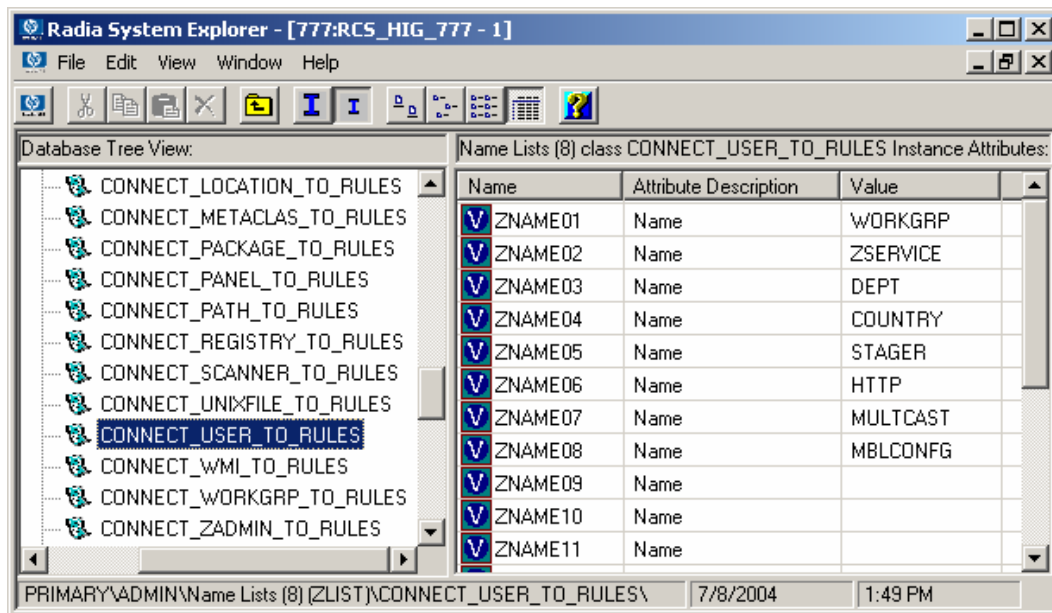


Figure 7.3 ~ The ZLIST.CONNECT_USER_TO_RULES instance.

An instance of this particular USER class can contain a connection to any of the following classes:

WORKGRP	ZSERVICE	DEPT	COUNTRY
STAGER	HTTP	MULTICAST	

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. Of course, if any class in your Radia Database needs to be able to connect to more than 20 different classes, you can edit the ZLIST class template and add additional ZNAME $_{nn}$ variables as needed.

Once the rules for connection have been established 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. You will then define the type of connection that can be made to the USER class by editing the rules variable.

When completed, drag-and-drop connection can be made to the new class. Figure 7.4 shows the connectable classes for the POLICY domain where the USER class is located.

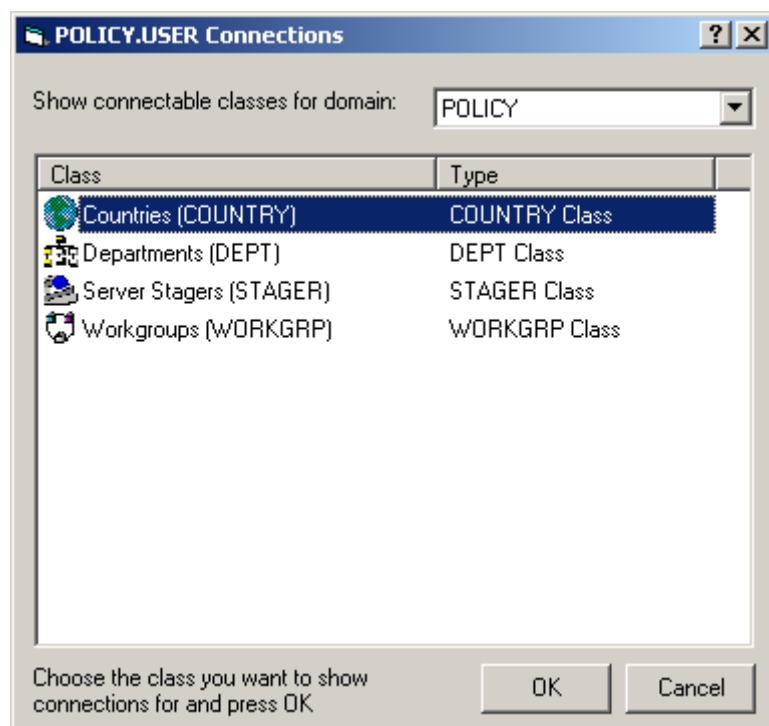


Figure 7.4 ~ The connectable classes for the POLICY domain.

Enabling Connections in Newly Added Classes

If you add a new class to the Radia Database that can validly contain connections to other classes you'll 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's variable in the MASTER_CONNECTION_RULES.

For example, the steps below show how to add the connections for a new class called TEST.

First, create an instance of the ZLIST class named `CONNECT_newclass_TO_RULES` where *newclass* is the name of the newly added class.

1. If you have not already done so, access the Radia System Explorer and navigate to the **PRIMARY.ADMIN.Name Lists (ZLIST)** class.
2. Right-click on ZLIST and select **New Instance**. Name your new instance. For example, let's create a rule for a new class called `CONNECT_TEST_TO RULES`.

Figure 7.5 on page 217 shows the added `CONNECT_TEST_TO_RULES` in the ZLIST class.

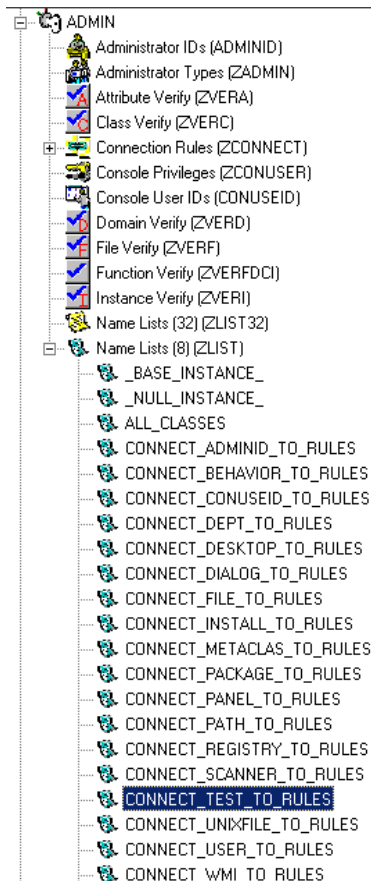


Figure 7.5 ~ The `CONNECT_TEST_TO_RULES` in the `ZLIST` class.

3. Next, edit the `CONNECT_newclass_TO_RULES` instance, setting each `ZNAMEnn` variable to the name of a class to which the newly added class can contain connections.

For example, we set the `ZNAME01` variable to allow connections to the `ZSERVICE` and to the `WORKGRP` classes.

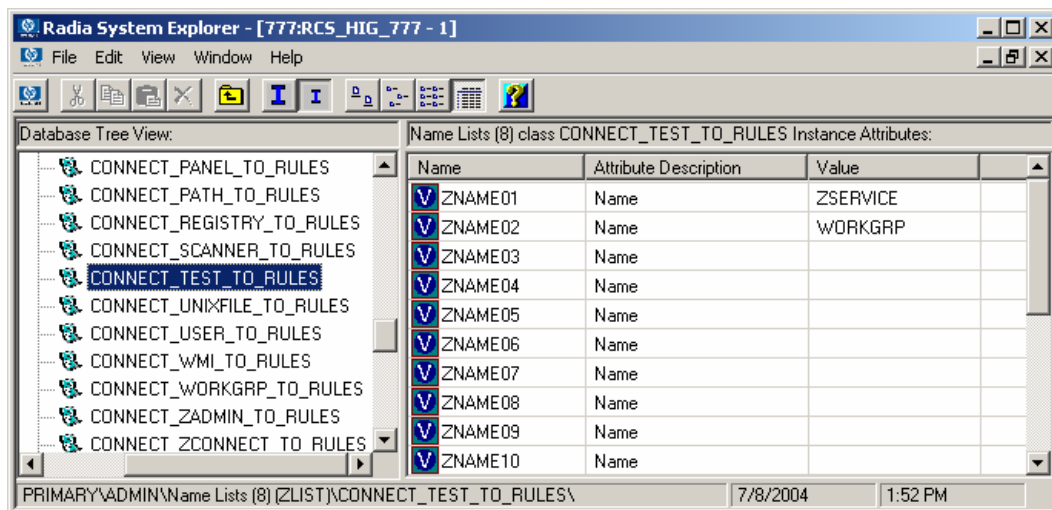


Figure 7.6 ~ The connectable classes for our TEST class.

Now you will need to add the connection rule variable to the Connection Rules (ZCONNECT) class.

4. Navigate to the **PRIMARY.ADMIN.ZCONNECT** class.
5. Right-click on the ZCONNECT class and select **Edit Class** from the context menu.
6. Add the new class variable to the class template for the ZCONNECT class.

For our example, we are adding the TEST attribute to the class. The length of the attribute will be 50, and the description is Test Connect List.

For information on editing a class, see *Editing a Class* beginning on page 61.

Note

When adding an attribute to the ZCONNECT Class, ensure that you change the attribute type to Class Connection. See Figure 7.7 on page 219.

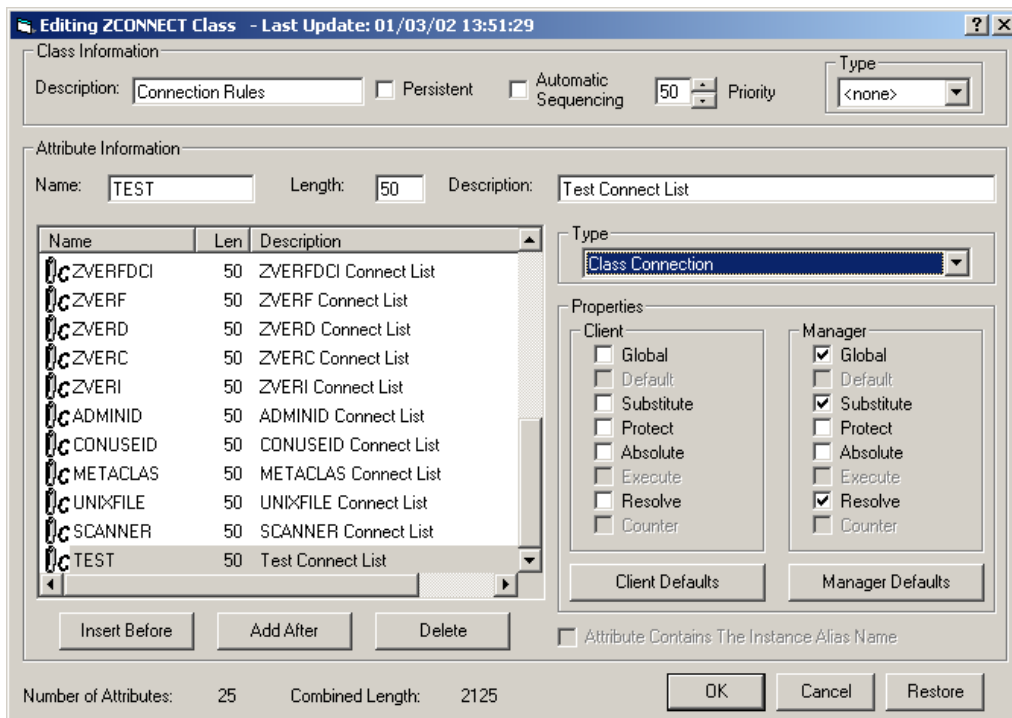


Figure 7.7 ~ Add the new class's name to the list of variables in the Connection Rules class.

Once you've added the new attribute to the class template, you'll need to set the value of the new attribute in the ZCONNECT.MASTER_CONNECTION_RULES instance.

7. Locate the new instance in the list view (right side of the screen). Set the newly added variable's value to ZLIST.CONNECT_newclass_TO_RULES where *newclass* is the name of the newly added class

For information on how to edit an instance, see *Editing an Instance* on page 75.

For example, see the value that has been set for the TEST attribute in Figure 7.8 on page 220.

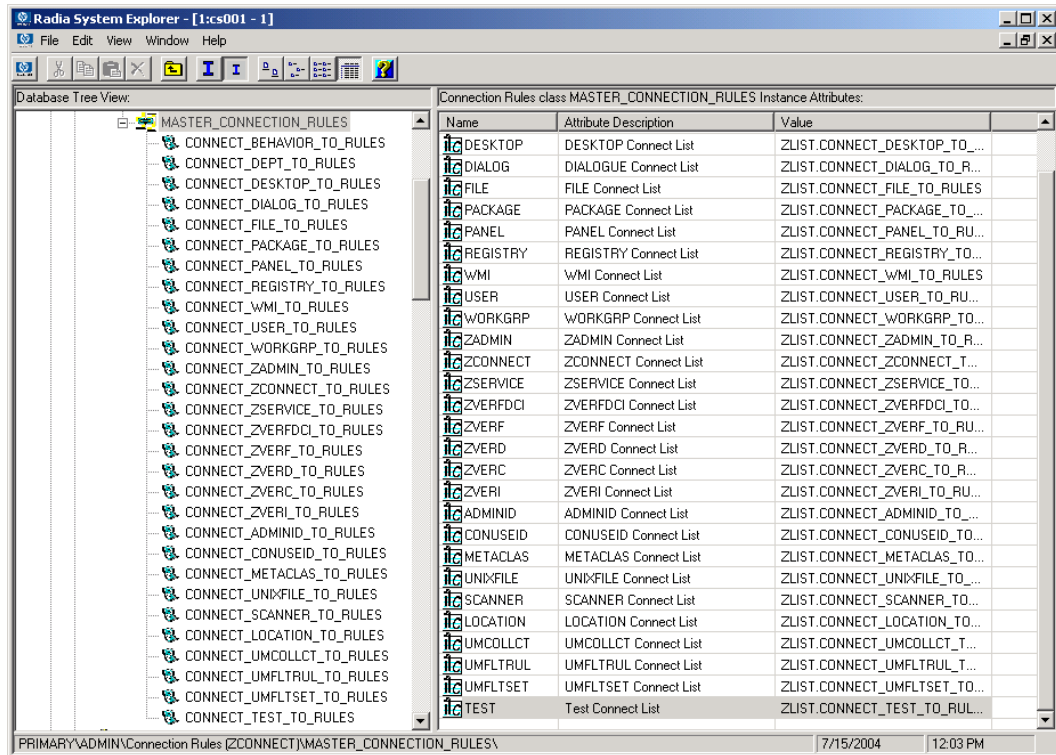


Figure 7.8 ~ Set the variable in the connection rule.

The new connection rule has been added to the MASTER_CONNECTION_RULES allowing drag-and-drop connections for the new class.

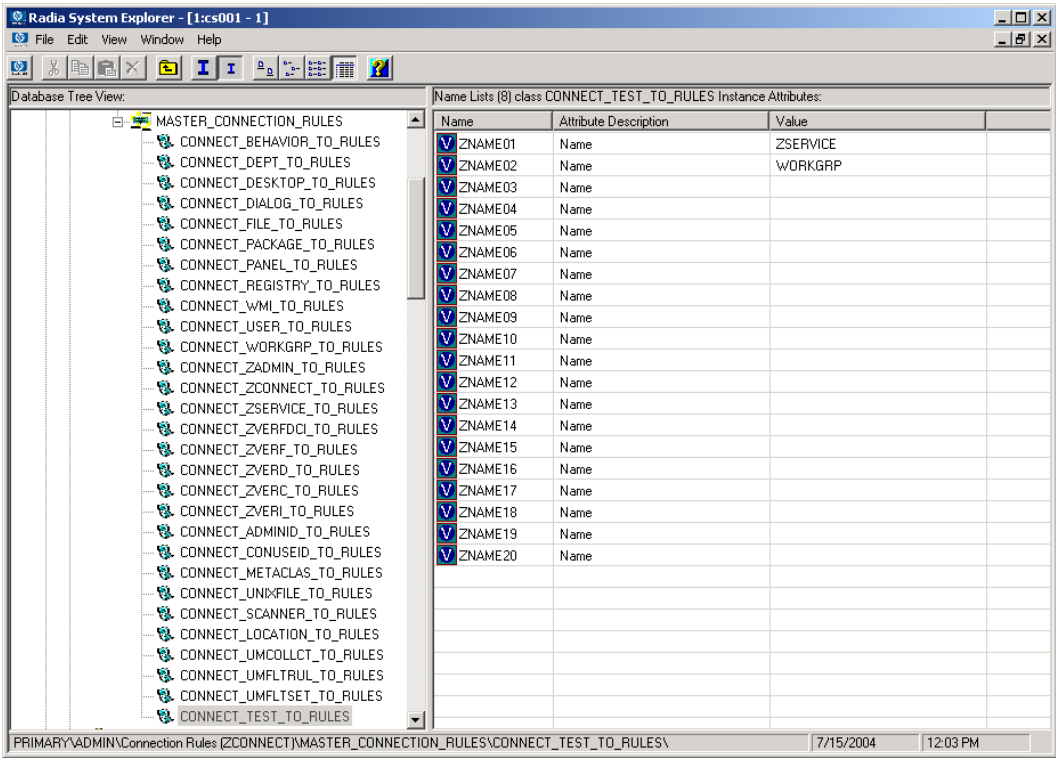


Figure 7.9 ~ The completed connection rule for our TEST class in the MASTER_CONNECTION_RULES.

Summary

- In order to allow for drag-and-drop capabilities for an added class, it is necessary to establish rules for the new class to follow.
- These connection rules are stored in the PRIMARY.ADMIN domain in the MASTER_CONNECTION_RULES instance of the Connection Rules (ZCONNECT) class.
- The Name List (ZLIST) class contains *all* of the rules for *all* of the classes.
- The ZCONNECT class houses the rules to enable connections between classes.
- Radia System Explorer then uses these rules to populate the class list in the Show Connections dialog box, and to control drag-and-drop connections.

Feature Set Editor

At the end of this chapter, you will:

- Know what the Radia Feature Set Editor is and what it does.
- Know how to access the editor.
- Understand the features of the editor.
- Know how manage feature sets in native MSI mode, if desired.

Feature Set Editor Overview

The Radia Feature Set Editor, a wizard type interface, guides you through the selection and configuration of the features of a Windows Installer-enabled software package managed by Radia. This allows you to use the features of Windows Installer while taking advantage of the packaging and deployment technologies available in Radia.

With the release of Office 2000, Microsoft introduced new technology for installing and maintaining software products on the Windows platform: Microsoft Windows Installer. Radia products can install and manage Windows Installer-enabled applications as follows:

- The Radia Publisher allows for Basic and Advanced methods of publishing and configuring Windows Installer-enabled Applications.

Note

For information regarding the packaging of Windows Installer-enabled applications, see the *Radia Extensions for Windows Installer Guide*. For information regarding the publishing of Windows Installer-enabled applications, see the *Radia Publisher Guide*. Both guides are available from the HP OpenView Support web site.

- The Radia Database enables HP products to store Windows Installer feature sets.
- Radia provides policy administration and state data gathering.
- Radia provides the ability to control the customization of MSI packages without the need to create Windows Installer Transform (MST) files.
- Radia provides the ability to discover previously installed, Windows Installer-enabled applications that were not deployed using Radia, and manage these packages.

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 wish to edit in the Radia Database.
2. Double-click this instance to display its MSI and MSIFEATS connections.
3. Right-click the MSIFEATS connection in the tree view, as follows:

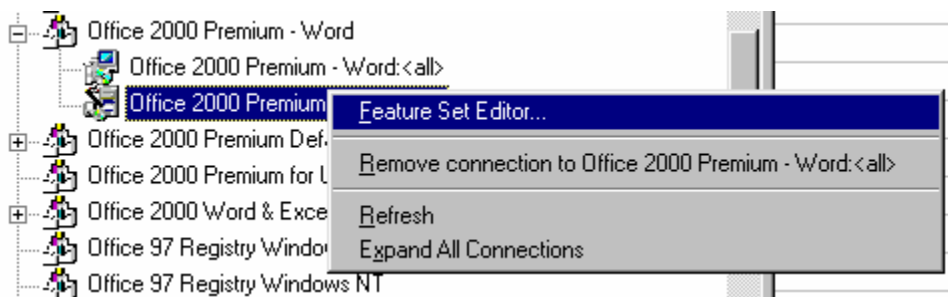


Figure 8.1 ~ Accessing the Feature Set Editor thru the Radia System Explorer tree view.

4. Select **Feature Set Editor** from the shortcut menu.

The **Feature Set Editor** dialog opens.

About the Radia MSI Feature Set Editor

As shown in Figure 8.2 below, the Feature Set Editor interface is divided into three areas, the **tree view** on the left, a **button list** on the right, and **package size information** along the bottom.

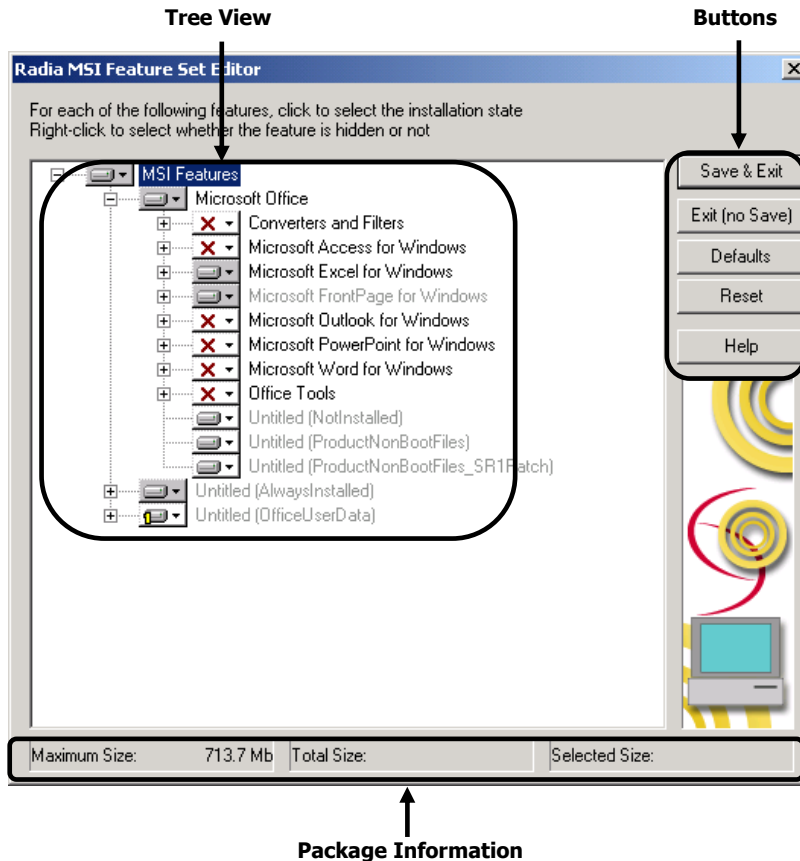


Figure 8.2 ~ The Feature Set Editor dialog box.

Tree View

Features are grouped into a hierarchy that you navigate using the tree control. Higher-level features consist of two or more lower-level (i.e., child) features.

To view the child features comprising any particular feature, click on the small plus sign in the box next to the feature. The plus sign changes to a minus sign, and the next lower level features are displayed beneath the selected feature in the tree view, as follows:

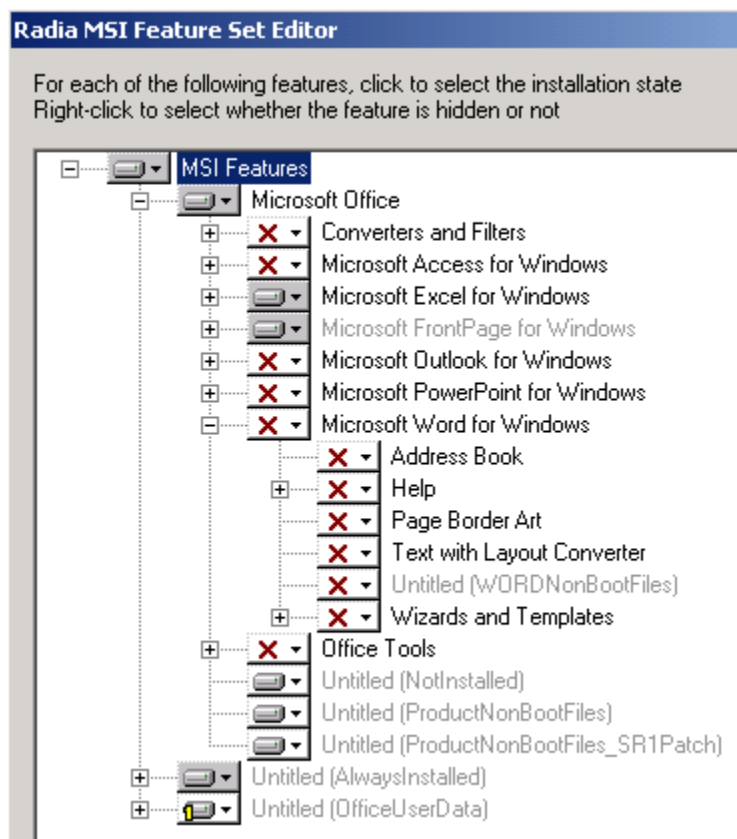


Figure 8.3 ~ Child features displayed.

To close the display of child features beneath a particular feature, click on the minus sign in the small box next to the selected feature:

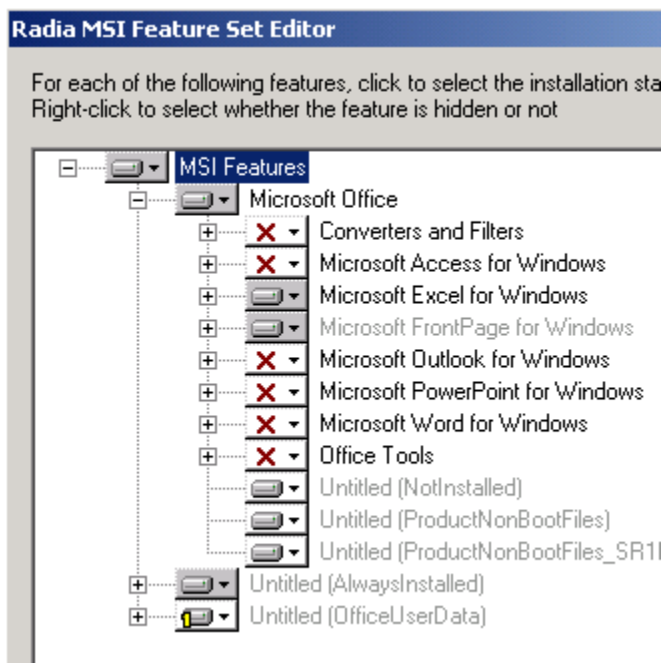


Figure 8.4 ~ Child features closed under Microsoft Word.

Features Menu

When you click on the drop-down button next to a feature, the **Features** menu opens, as shown here.

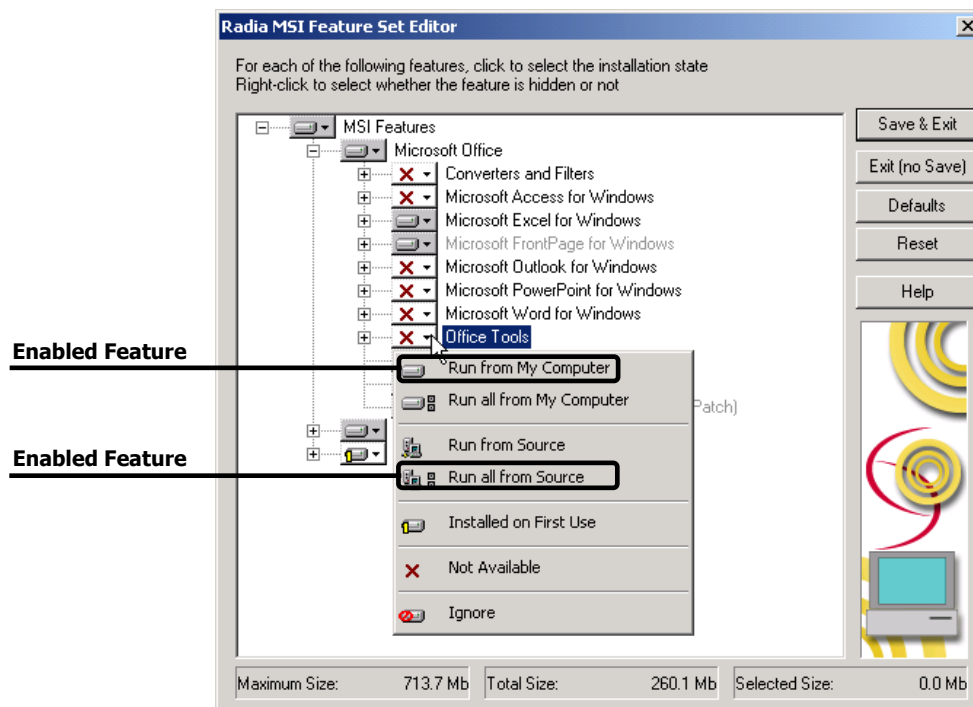


Figure 8.5 ~ The Features menu opens when the drop-down arrow next to a feature is clicked.

The graphic on the button that appears next to each feature indicates the current attribute assigned to the feature. And, each of the features has an attribute that indicates how that feature is to be installed and maintained.

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 explanation of each attribute for information about how the attribute is propagated to child features in Table 8.1 beginning on page 230.

Table 8.1 ~ Features Menu Options




Feature Icon and Name	Description of Function
 Run from My Computer	<p>Features with the Run from My Computer attribute are physically installed on a hard drive on the client computer. Features with this attribute run with the best performance, because they are run from the end user's own machine.</p> <p>When you select this attribute for a feature, the Run from My Computer attribute is applied automatically to all features that are children of the selected feature, unless the child feature's attribute is already set to Installed on First Use or Not Available.</p> <p>The selected feature's children are those features that appear on branches of the tree, connected to and beneath the selected feature's branch of the tree.</p>
 Run all from My Computer	<p>Features with the Run all from My Computer attribute are physically installed on a hard drive on the client computer. All features that are children of the selected feature have their attributes changed to Run from My Computer, and they, too, are physically installed on the client computer.</p> <p>Features with this attribute run with the best performance, because they are run from the end user's own machine. Mobile computer users should have all of the features they require while traveling set to Run from My Computer or Run all from My Computer. Otherwise, when they go to use a feature that is not installed on their computer's hard drive, they will be prompted for the original installation media, which may not be available at that time.</p>
 Run from Source	<p>Features with the Run from Source attribute are not physically installed on a hard disk on the client computer, rather they are run from the Administrative Installation Point (AIP). This is normally a network share persistently accessible to the client computer.</p> <p>When you select this attribute for a feature, the Run from Source attribute is applied automatically to all features that are children of the selected feature, 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 requirement of having an active connection to the Administrative Installation Point.</p>

Table 8.1 ~ Features Menu Options





Feature Icon and Name	Description of Function
 Run all from Source	<p>Features with the Run all from Source attribute are not physically installed on a hard disk on the client computer, rather they are run from the Administrative Installation Point (AIP). This is normally a network share persistently accessible to the client computer.</p> <p>When you select this attribute for a feature, the Run from Source attribute is applied automatically to all features that are children of the selected feature, 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 requirement of having an active connection to the Administrative Installation Point.</p>
 Installed on First Use	<p>Features with the Installed on First Use attribute are not installed on the client 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 required 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 client computer.</p> <p>In order to install such a feature, the client computer must be running the Radia Redirector service, and it must be able to connect to the Radia Configuration Server computer.</p> <p>Using this attribute minimizes the use of hard disk space on client computers by allocating space only to features that they actually use. On the other hand, end users will be subject to the delay required to install the feature at the time when the first invoke it, which may surprise and displease them.</p> <p>When you select this attribute for a feature, the Installed on First Use attribute is applied automatically to all features that are children of the selected feature, unless the child feature's attribute is already set to Not Available.</p> <p>See note on page 233 for additional information.</p>

Table 8.1 ~ Features Menu Options

Feature Icon and Name	Description of Function
 Not Available	<p>Features set to Not Available are not installed, are unavailable to the end user and may affect other services. The attributes for all child features are also set to Not Available.</p> <p>For example, one service can be defined that deploys Microsoft Word, and all other features of Office are set to Not Available. Another service can be defined that deploys Microsoft Excel, and all other features of Office are set to Not Available.</p> <p>Now, imagine the end user installs the Microsoft 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 Microsoft Excel service indicates that all other features of Office are Not Available. Therefore, Microsoft Excel is installed <i>and Microsoft Word is uninstalled</i>.</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 HP-managed generally don't 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>
 Ignore	<p>Note: The Radia feature state of IGNORE will be maintained for backward compatibility. You will no longer be responsible for determining which features need to be ignored by a service.</p> <p>Features set to Ignore are not installed and are unavailable to the end user, but will not affect other services. Radia 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 can be defined that deploys only Microsoft Word, with all other features of Office set to Ignore. Another service can be defined deploying Microsoft 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 Radia Database. The package is cloned in the Radia Database by copying it with another name. You can then use the Feature Set Editor to edit the features of each package.</p>

Important note from Microsoft regarding Office 2000 installation

"*Installed on First Use* is supported only on operating systems that support Windows Installer shortcuts, including Windows 98; Windows 95 with Internet Explorer 4.01 with Service Pack 1 or later; and Windows NT 4.0 with Service Pack 3 and Internet Explorer 4.01 with Service Pack 1 or later. On Windows 95 and Windows NT 4.0, you must also have Active Desktop installed (but not necessarily enabled) before you install Office. If you install Office on an operating system that does not support Windows Installer shortcuts, then features set to *Installed on First Use* are set to *Run from My Computer* instead."

Current Feature Settings

The current setting for each feature is graphically indicated on the button next to the feature's name. The graphic on the button indicating the current feature setting matches the graphic on the matching choice from the shortcut menu that opens when you click the button:

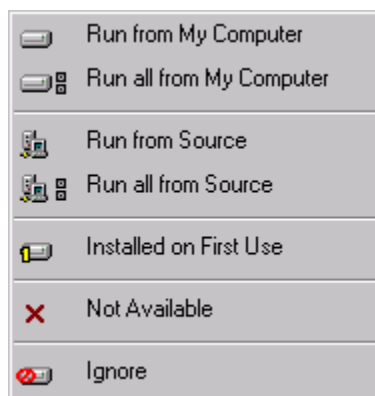


Figure 8.6 ~ The graphic settings for features.

This can be seen in Figure 8.6 above. The Microsoft Access for Windows feature button shows the graphic indicating that it is **Not Available**, while the **Text with Layout Converter** feature shows the graphic indicating that the feature is **Installed on First Use**.

Names of features whose attributes have been changed during the current session of the Radia MSI Feature Set Editor appear in boldface in the tree:

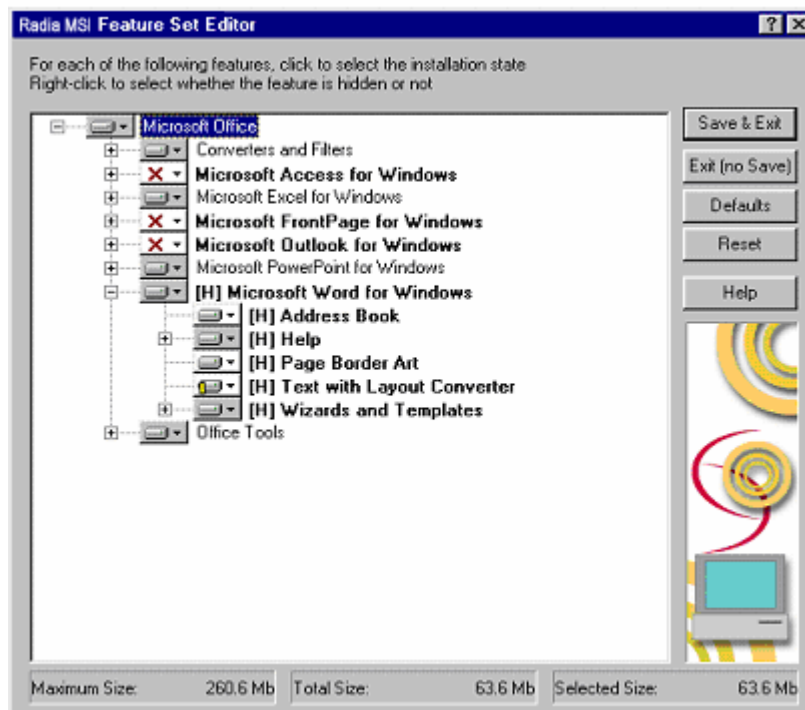


Figure 8.7 ~ Names of features appearing in boldface indicate a change to their attributes has occurred.

In Figure 8.7 above, Access, FrontPage, and Outlook have been marked **Not Available**, and Word has had all of its features hidden during the current Radia MSI Feature Set Editor session.

Hide and Disable Features

When you right-click on the name of a feature (in the following example, Microsoft Word for Windows), the shortcut menu for **Hide** and **Disable** pops up:

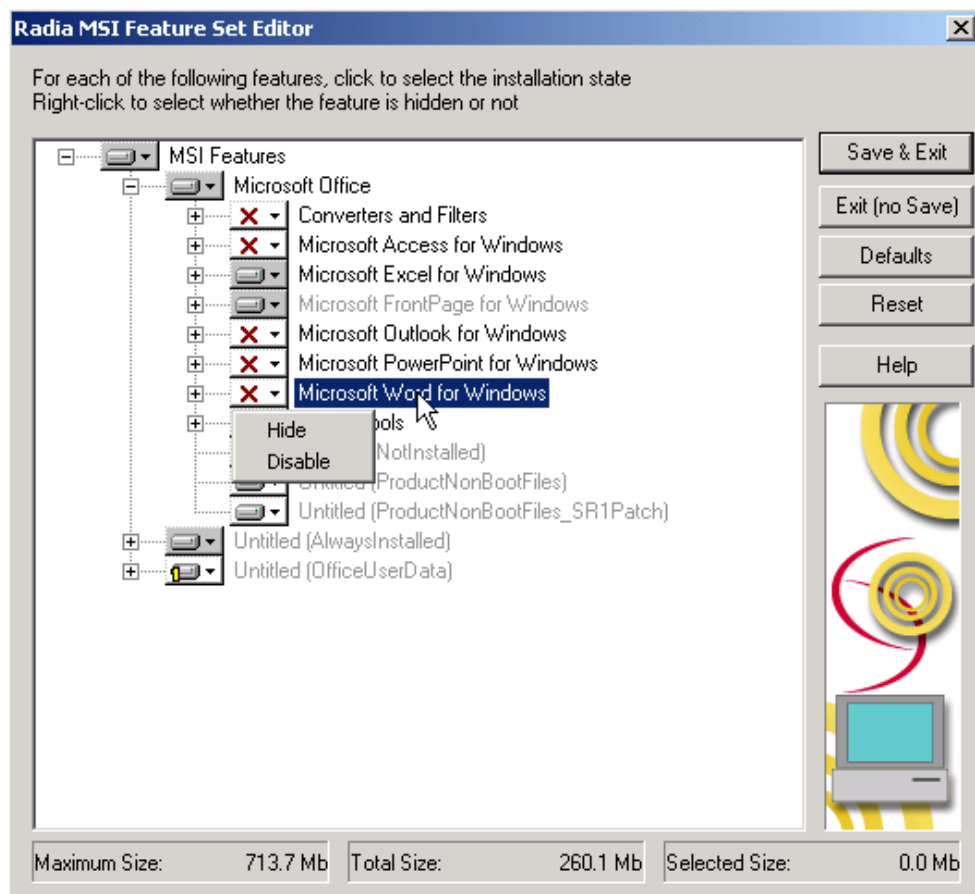


Figure 8.8 ~ A shortcut menu for **Hide** and **Disable** opens when a feature's name is right-clicked.

When you select **Hide** from the shortcut menu, the selected feature and all of its child features are prevented from appearing 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 Radia MSI Feature Set Editor tree with the prefix **(H)** before the name of the feature, as follows:

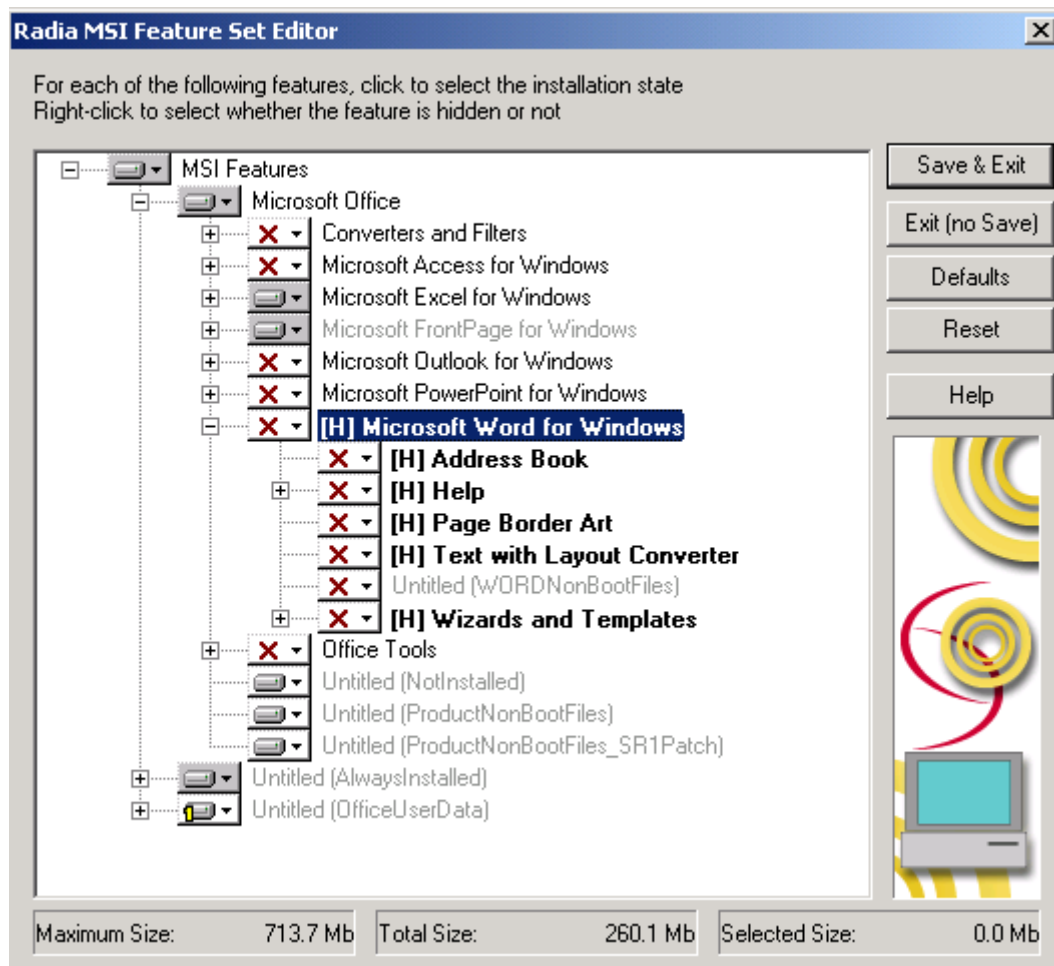


Figure 8.9 ~ All hidden features appear with the prefix (H) before their names.

When you select **Disable** from the shortcut menu, the selected feature and all of its child features are ignored by the Radia client. A disabled feature's value is determined by Windows Installer, from the combination of MSI and MST files, instead of the value in the Radia instance.

All features that are disabled appear in the Radia MSI Feature Set Editor tree with the prefix **(D)** before the name of the feature, as follows:

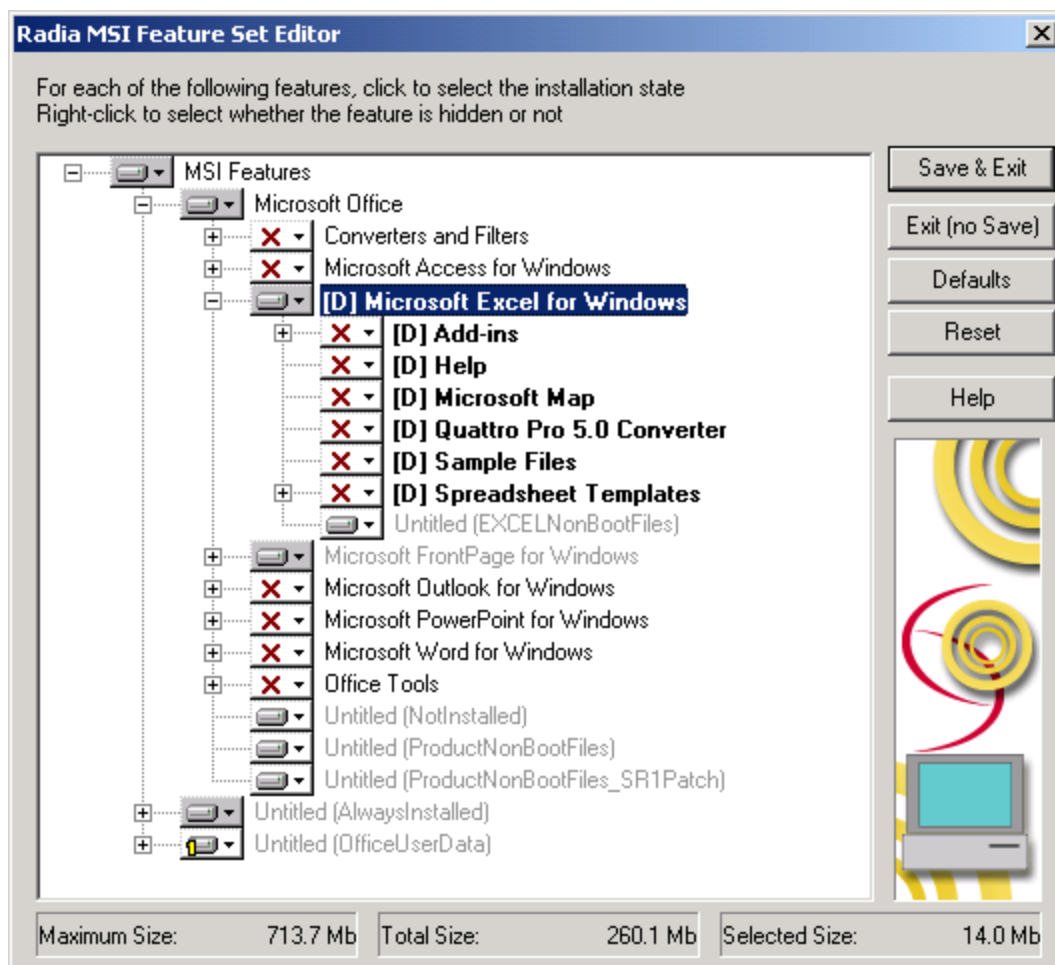

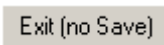

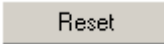
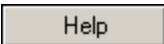


Figure 8.10 ~ All disabled features appear with the prefix (D) before their names.

Feature Set Editor Buttons

The Feature Set Editor buttons are located on the right side of the Feature Set Editor window. See Table 8.2 on page 238 for the actions invoked by clicking these buttons.

Table 8.2 ~ 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 to 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 Exit (no Save) to discard any changes that may have been made and to 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 contained in the MSI file distributed with the software by the software vendor. The default setting will not be permanently applied to the feature set until the feature set is saved by clicking Save & Exit .
	Click Reset to discard any changes made to feature settings since opening the current session of the Radia MSI Feature Set Editor. Settings for all features are refreshed by re-reading them from the Radia Database.
	Click Help to access the Radia System Explorer Help topics. The screen will automatically open to the Feature Set Editor chapter in Help.

Package Information

Use the package information to view the size of the current Windows Installer-enabled features that you are configuring.

Maximum Size

The number shown is the total number of bytes that would be required on the client computer to install all features in the entire feature tree, if they were all given the **Run from My Computer** or **Run All from My Computer** attribute.

Total Size

The number shown is the total number of bytes required on the client computer to install all features in the entire feature tree that have the **Run from My Computer** or **Run All from My Computer** attribute.

Selected Size

The number shown is the total number of bytes required on the client computer to install the currently selected (i.e., highlighted) feature and its child features. The size calculation includes only features with the **Run from My Computer** or **Run All from My Computer** attribute.

Windows Add/Remove Programs Applet

When a Windows Installer-enabled software package is installed on a client computer, it is normally added to the list of applications that can be manipulated by the Add/Remove Programs applet of the Control Panel.

Radia products give you the option to avoid listing the Windows Installer –enabled software package in the Add/Remove Programs applet of the Windows 9X, NT, 2000 and XP products. This prevents the end user from interfering with Radia products' centralized management of the application. Alternatively, you can selectively hide individual features from access via the Add/Remove Programs applet.

To completely avoid listing the application in the Add/Remove Programs applet, use Radia System Explorer 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.

Change the value of the SHOWINCP variable to N, as follows:

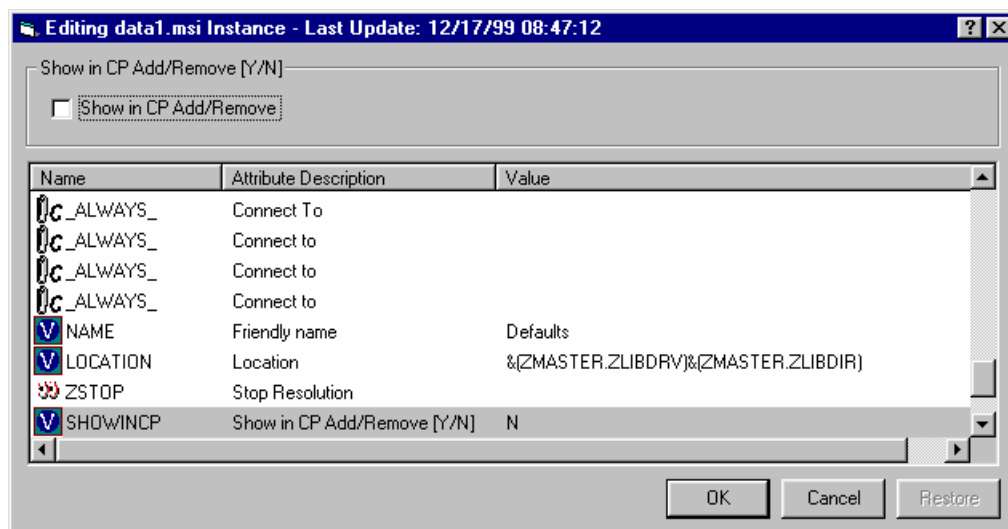


Figure 8.11 ~ The Editing DATA1.MSI Instance dialog box is used to change the value of SHOWINCP to N.

When the end user selects the Windows Installer-enabled software package in the Add/Remove Programs applet and clicks **Add/Remove**, Microsoft Windows Installer is invoked, and presents its native user interface for manipulating features of the installation on the client computer.

If the end user clicks **Add or Remove Features**, Microsoft Windows Installer displays its **Update Features** windows.

The Radia MSI Feature Set Editor's **Hide** function enables the administrator to prevent features from appearing in the Microsoft Windows Installer **Update Features** dialog box. This enables you to maintain control over the end user's installation of the application software's features via your Radia management product's capabilities, and to avoid possible end-user interference.

Copying a Windows Installer Enabled Package

One of the key advantages of using Radia management products to manage Windows Installer-enabled applications is the ability to easily specify variants of the software package, and make those variants available to targeted populations of users.

Using the advanced capabilities of Radia, you need only create one comprehensive Administrative Installation Point, and promote it to the Radia Database. All variant sets of features are encapsulated in individual packages in the Radia Database.

To make a new variant set of features

Clone 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 resulting shortcut menu.

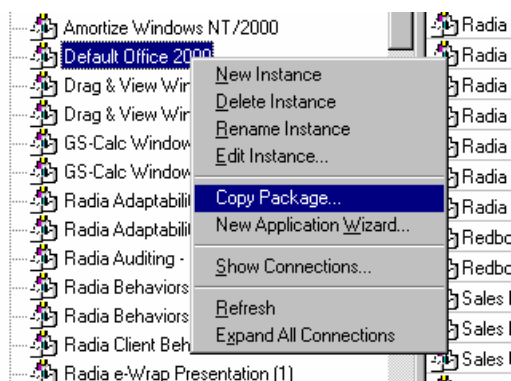


Figure 8.12 ~ Copying a package.

The following dialog box opens.

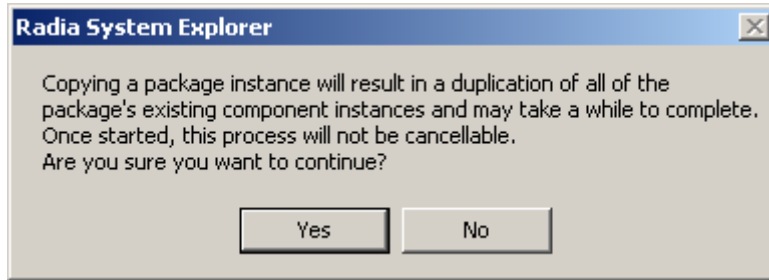


Figure 8.13 ~ Copy Package dialog box.

2. Select **Yes** to continue the Copy Package function. Select **No** to cancel the Copy Package function. For our example, we selected Yes.

Copying the Package

Copying the package copies the package instance and all of the component class instances that belong to the package instance. The resulting package is a complete clone of the original. This operation maintains the integrity of the Radia Database, because each package retains control over its own components.

3. Indicate the names for the newer copied package:

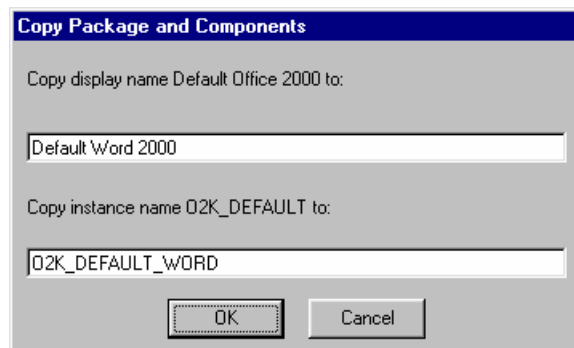


Figure 8.14 ~ Copy Package and Components dialog box.

4. Click **OK** to continue. The Radia Configuration Server takes a few moments to clone the package in the Radia Database.

5. When the newly cloned package appears in the Radia System Explorer's tree view, right-click on the MSIFEATS instance beneath the package in the tree view, and select **Feature Set Editor** from the shortcut menu:

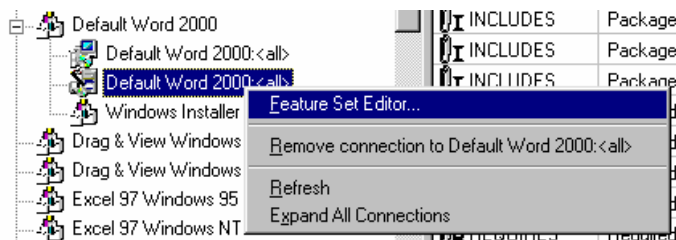


Figure 8.15 ~ Opening the Feature Set Editor from the shortcut menu.

The Radia MSI Feature Set Editor opens.

6. Set the properties of the applications that are *not* to be deployed to **Ignore**:

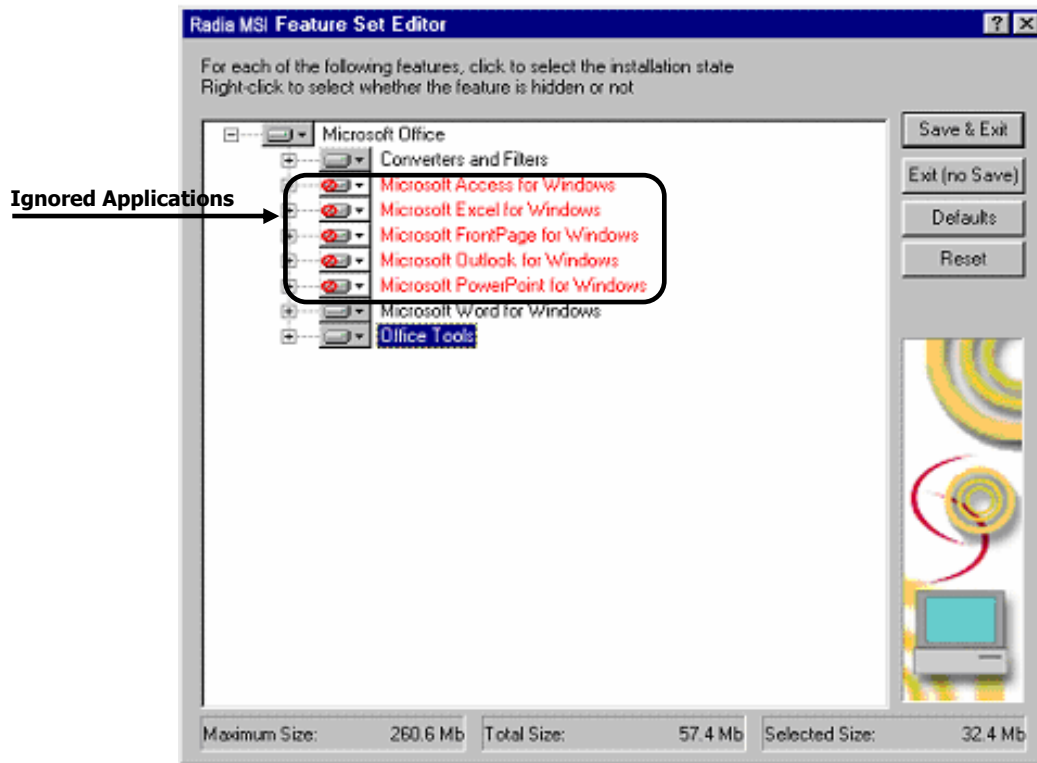


Figure 8.16 ~ The Feature Set Editor displaying applications whose properties are set to Ignored.

7. Click **Save & Exit**, and the feature set for the package will be 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 Radia are two separate functions. You may *not* want to use the Radia MSI Feature Set editor to edit the transform (MST) files. To manage feature sets in native MSI mode, publish the package and MST files using the following procedure:

1. Use the Radia Publisher in Advanced mode to create the ACP and publish the Windows Installer Package.
2. After completing the publishing process, use Radia System Explorer to remove the connection to SOFTWARE.MSIFEATS.packageprefix_* from the Application Packages (PACKAGE) instance.
3. If you have multiple MST files, use the Radia Packager in Component Selection mode to promote each of the MST files into individual packages. This will create a new PACKAGE instance for each MST file.
4. Connect the MSI PACKAGE instance to each of the transform file's PACKAGE instances.
5. Create one Application (ZSERVICE) instance for each of the transform packages.
6. Assign subscribers to the appropriate transform package's service.

Caution

Any changes to or replacement of the *.MST file will cause Radia to call MSIEXEC to uninstall and reinstall the application.

When the service is installed or modified, Radia will detect the absence of the MSIFEATS object, and will use the native MSI mode, MSIEXEC. Radia will call MSIEXEC natively with the proper location of the data source. If Radia 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 Radia calls MSIEXEC and detects more than one MST file, it will run MSIEXEC for the MSI file only. The MSI native behavior will be used for verify, repair, and update.

Caution

If you are using an MST file to include a new feature that is not already a feature available in the Radia Feature Set Editor (MSIFEATS), then 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, then use the procedure outlined in this section.

Summary

- Use the Feature Set Editor to select and configure each of the features of a Windows Installer-enabled software package.
- Features are grouped in a hierarchy. High-level features consist of two or more lower level features.
- Each of the features has an attribute that indicates how that feature is to be installed and maintained.
- Alternative procedures allow you to manage feature sets in native MSI mode, if desired.

Registry File Editor

At the end of this chapter, you will:

- Understand the Registry File Editor.
- Know how to edit data in a registry key.

Use the Registry File Editor to edit the properties or values for registry keys associated with a Radia package.

The Registry File Editor

When using Radia 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. The **Registry File Editor** allows you to modify this file. Then, the file is sent (or re-promoted) to the Radia Configuration Server and the instance in the Radia Database is updated.

See the *Radia Application Manager Guide* for information on creating and promoting packages.

Accessing the Registry File Editor

To access the Registry File Editor

1. Locate the Registry Resource instance that represents the software package whose registry settings you wish to change in the Radia Database.
2. Right-click the instance in the tree view. A context menu opens. For example, locate the registry instance for the Amortize Windows package and right-click on it.

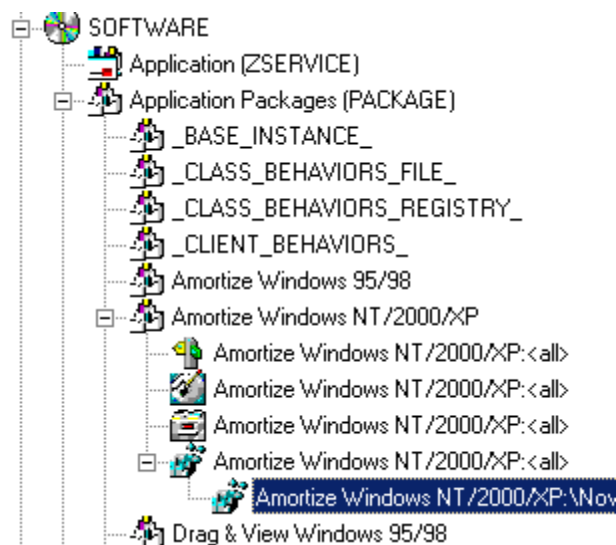


Figure 9.1 ~ Accessing Registry File Editor.

3. Select Edit Registry Resource.

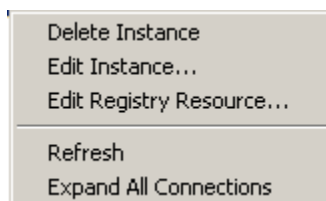


Figure 9.2 ~ Select Edit Registry Resource from the context menu.

The **Registry File Editor** dialog box opens.

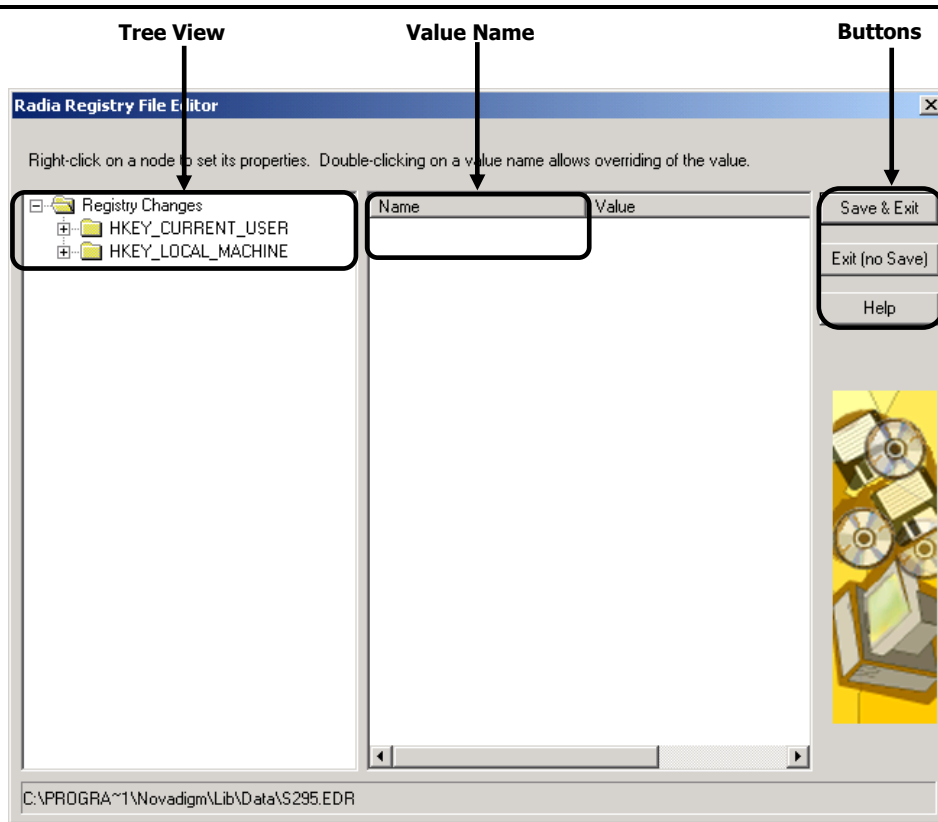


Figure 9.3 ~ Registry File Editor.

- 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.

Expanding and Collapsing the Tree

Nodes are grouped into a hierarchy. When you set a property to a node, the Registry File Editor applies changes to all nodes within that node's branch in the tree view.

The branches of the tree view can be expanded to view the nodes.

To expand a single node in the tree view

- Click the plus sign in the small box next to the node.

The plus sign changes to a minus sign and the next level opens beneath the selected node in the tree view.

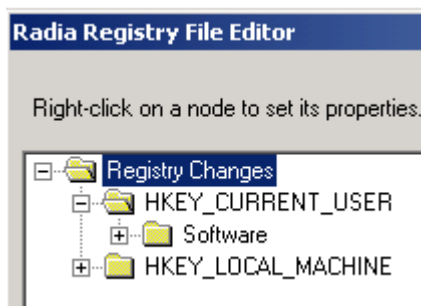


Figure 9.4 ~ Expand a node.

To collapse a single node on the tree

- Click the minus sign in the small box next to the node. The minus sign changes back to a plus sign.

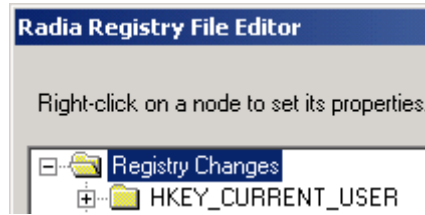


Figure 9.5 ~ Collapse a node.

Set Properties Context Menu

Right-click on any node of the tree view in the Radia Registry File Editor. A context menu opens as shown in Figure 9.6.

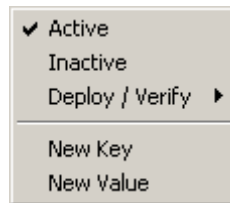



Figure 9.6 ~ Setting node properties.

For specific information regarding the options available through this context menu, see Table 9.1 beginning on page 255.

Table 9.1 ~ Registry File Editor Context Menu Options

Action	Description
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	<p>Click Deploy/Verify to access the following submenu:</p> <ul style="list-style-type: none"> • Unconditional/Enforced Radia deploys the registry key every time the subscriber connects to the Radia 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 Radia 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 Radia deploys the registry key only if it doesn't 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 on New Key if you would like to add a new value to your registry. Clicking New Key will display a dialog box to enter the New Key name. Click OK to accept the new key, or Cancel to return to the Radia Registry File Editor.
New Value	<p>Click on New Value to add a value to a newly created key or to add a value to an existing registry key. Clicking on New Value opens a dialog box where you will be required to enter the following information:</p> <p>Value Name Enter the name of the value you will be adding to the registry file. This name will appear in the Value Name (center) of the Radia Registry File Editor window.</p> <p>Data Type Use the drop-down menu to select from the list of available data types. The type selected is as follows:</p> <ul style="list-style-type: none"> • REG_SZ A text sequence most typically used for descriptive information. This is the default setting. • REG_DWORD Is data represented by an integer that is 4 bytes long and is displayed in the binary, hexadecimal, or decimal format. • REG_EXPAND_SZ Is 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.

Editing a Registry Key

Editing a registry key's value is accomplished by accessing the key through the Radia Registry File Editor. Follow the example provided below to edit the information in a registry key.

Caution

Extreme caution must be used when using the Radia Registry File Editor. Editing registry keys without thorough understanding of their functions may render a package or service useless.

To edit the data in a registry key

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**. The **Radia System Explorer Security Information** dialog box appears.

Note

The **User ID**, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. Check with your Radia security administrator to obtain your own **User ID** and **Password**, if necessary.

2. If necessary, type a **User ID** and **Password**, then click **OK**. The **Radia System Explorer** splash screen appears briefly.
3. Double-click **PRIMARY**.
4. Expand the **SOFTWARE** domain and then the **Application packages (PACKAGE)** Class.
5. Expand the Amortize Windows NT/2000/XP package.
6. Expand the Registry instance as shown in Figure 9.7 on page 257.

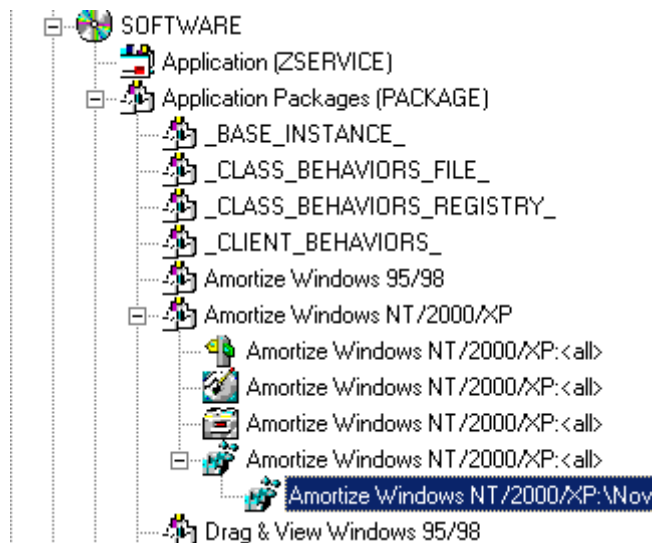


Figure 9.7 ~ Select the Registry Instance of the package.

7. Right-click the registry file and select **Edit Registry Resource** from the context menu. The **Radia 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.

Double click the registry key in the list view.

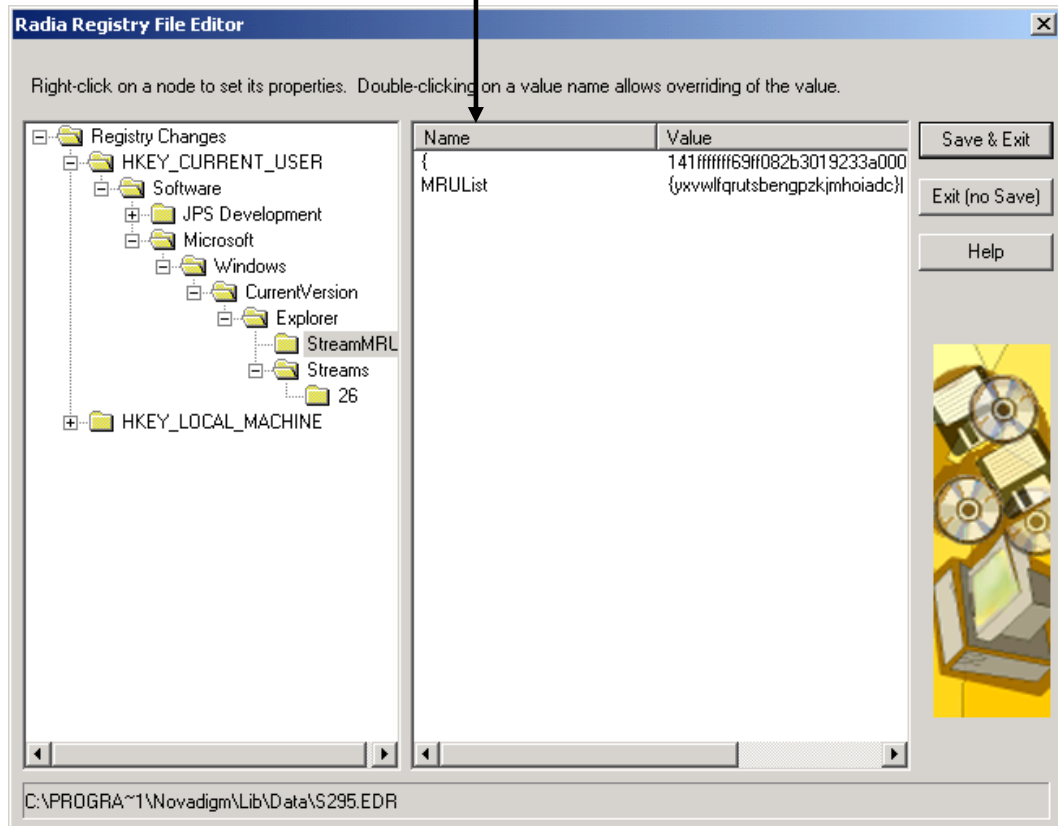


Figure 9.8 ~ Select the key you are editing by double-clicking on the value name in the list view.

The **Override Value** dialog box opens.

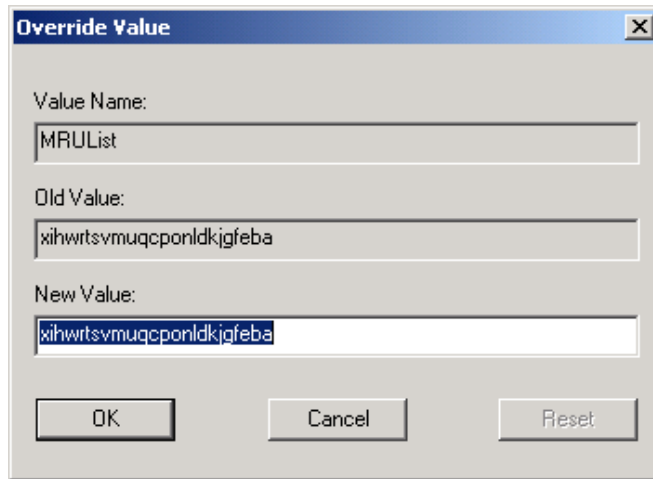


Figure 9.9 ~ Override Value dialog box.

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.

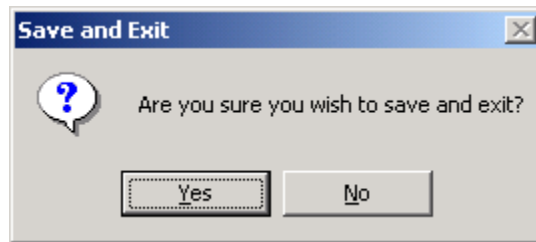


Figure 9.10 ~ Exiting the Registry dialog box.

OR

Click **No** to return to the **Registry File Editor**.

Summary

- Use the Registry File Editor to edit the properties or values for registry keys associated with a Radia package.
- The Registry File Editor applies changes to all nodes within that node's branch in the tree view.
- Once you have edited a registry key's data, you can restore it to its original value.

Support for the Radia Inventory Manager

At the end of this chapter, you will:

- Understand the audit domain.
- Understand WBEM and the Radia Client.
- Be familiar with the Radia Integration Server.

About the Radia Inventory Manager

The Radia Inventory Manager client is a utility used to discover configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results.

The Radia Inventory Manager client is used with the Radia Integration Server. The Radia Integration Server stores and maintains the discovery information to be viewed within an ODBC-compliant (Open Database Connectivity) database. The Radia Integration Server then provides the reports via a Web browser interface.

This manual explains how to install and use the Radia Inventory Manager client. Choose the appropriate strategies suited for your enterprise needs.

Overview

Systems administrators use the Radia System Explorer to manipulate the contents of the Radia Database. They specify what inventory management tasks to perform and on which client computers to perform them.

The collection of inventory information is performed on the Radia Inventory Manager computer when a subscriber connects to and installs the auditing software with the Radia Configuration Server as follows:

- **Software Manager** installs the software when the user visits his Radia Web page.
- **Application Manager** installs the software in one of the following ways:
 - through a log-on script
 - when the user double-clicks a desktop Radia Connect icon
 - according to a schedule
 - using the Notify capabilities of the Radia Configuration Server

The results of the inspection are then sent back to the Radia Configuration Server.

The Radia Configuration Server can store these inspection results in the Radia Database's PROFILE file and/or forward them to the Radia Integration Server for insertion into an ODBC-compliant database.

Any unwanted files discovered on a user's computer can be captured and/or deleted. This enables administrators to remove unauthorized content, such as games, from end users' computers.

Radia and WBEM

The Radia Inventory Management client queries the WBEM namespace (that is, the WBEM database) and sends the results back to the Radia Configuration Server. All information collected by WBEM is available to the Radia Inventory Manager client. The collected information is then stored in the Radia Integration Server.

For client computers with WBEM (Web-Based Enterprise Management) installed, the Radia Inventory Manager executes an HP-proprietary method (RIMWBEM) to query the WBEM namespace.

For client computers that do not have WBEM installed, the Radia Inventory Manager executes HP proprietary methods to *directly* inspect the hardware (built into the Radia Client – ZCONFIG) and/or the file system (RIMSFSCAN).

The Radia Integration Server

The Radia Integration Server is a Tool Command Language (Tcl) based Web server that can reside on the same computer as the Radia Configuration Server, or on a separate computer. It builds and updates a structured query language (SQL) database of your choice via ODBC.

The ODBC driver for the database you choose *must* be installed prior to installing the Radia Integration Server. Depending upon which database you choose, you may have to perform administrative tasks for allocating space and establish a user ID and password for the Radia Integration Server's use.

The Radia Integration Server will automatically create the necessary tables in the database you choose.

Summary

- The Radia Inventory Manager discovers configuration information on remote computers and enables centralized reporting and administration based on discovery results.
- All information collected by WBEM is available to the Radia Inventory Manager.
- The Radia Inventory Manager queries the WBEM namespace (that is, the WBEM database) and sends the results back to the Radia Configuration Server.
- Complete information concerning the Radia Inventory Manager, the AUDIT domain and the services provided through Radia, can be found on the HP OpenView web site.



Radia System Explorer Shortcut Menus

Radia System Explorer Shortcut Menus

In the Radia System Explorer you can add, copy, edit, rename, and delete Radia Database components (domains, classes, and instances). It also provides extensive support for making correct connections between instances.

You can access these functions from shortcut menus that open when you right-click targets in the Radia System Explorer window. The menu that opens contains different choices depending upon which target you right-click. The following table shows how the choices in the shortcut menus can differ based upon the target of the right-click.

Note

The **Prune Below** menu choice appears only when the tree has been opened below the target of the right-click.

Table A.1 ~ Primary, Domain and Class Shortcut Menus

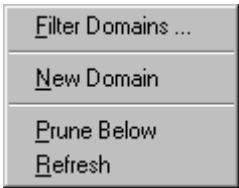

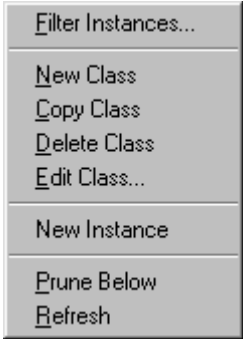
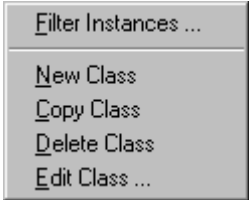
Target of Right-Click	Context Menu
<ul style="list-style-type: none"> PRIMARY file in the tree view 	 <p>A vertical context menu with four items: 'Filter Domains ...', 'New Domain', 'Prune Below', and 'Refresh'.</p>
<ul style="list-style-type: none"> Any domain in the tree view 	 <p>A vertical context menu with five items: 'Filter Classes ...', 'New Domain', 'New Class', 'Prune Below', and 'Refresh'.</p>
<ul style="list-style-type: none"> Any class in the tree view 	 <p>A vertical context menu with eight items: 'Filter Instances...', 'New Class', 'Copy Class', 'Delete Class', 'Edit Class...', 'New Instance', 'Prune Below', and 'Refresh'.</p>
<ul style="list-style-type: none"> Any class in the list view of the Radia System Explorer window 	 <p>A vertical context menu with five items: 'Filter Instances ...', 'New Class', 'Copy Class', 'Delete Class', and 'Edit Class ...'.</p>

Table A.2 ~ ADMIN Instance Context Menus	
Instance Name	Context Menu
<ul style="list-style-type: none">Administrator Ids (ADMINID)Administrator Types (ZADMIN)Class Verify (ZVERC)Connection Rules (ZCONNECT)Console user Ids (CONUSEID)Domain Verify (ZVERD)File Verify (ZVERF)Function Verify (ZVERFDCI)Instance Verify (ZVERI)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Show Connections...</div><div>Refresh</div><div>Expand All Connections</div></div>
<ul style="list-style-type: none">Attribute Verify (ZVERA)Console Privileges (ZCONUSER)Name Lists (32) (ZLIST32)Name Lists (8) (ZLIST)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Refresh</div><div>Expand All Connections</div></div>

Table A.3 ~ AUDIT Instance Context Menus	
Instance Name	Context Menu
<ul style="list-style-type: none">Application (ZSERVICE)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Optimize Service...</div><div>Build State File</div><div>Show Connections...</div><div>Notify Subscribers</div><div>Prune Below</div><div>Refresh</div><div>Expand All Connections</div></div>

Note: Build State File is available if the Radia Extensions for Windows Installer is installed.

Table A.3 ~ AUDIT Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none"> Audit Packages (PACKAGE) 	<div> <div>New Instance</div> <div>Delete Instance</div> <div>Rename Instance</div> <div>Edit Instance...</div> <div>Add Components...</div> </div> <div> <div>Copy Package...</div> <div>New Application Wizard...</div> </div> <div> <div>Show Connections...</div> </div> <div> <div>Refresh</div> <div>Expand All Connections</div> </div>
<ul style="list-style-type: none"> Behavior Services (BEHAVIOR) Desktop (DESKTOP) File (FILE) File Scanner (FILESCAN) File Scanner Filters (FILTER) Inventory Scanners (SCANNER) Path (PATH) WBEM (WBEM) 	<div> <div>New Instance</div> <div>Copy Instance</div> <div>Delete Instance</div> <div>Rename Instance</div> <div>Edit Instance...</div> </div> <div> <div>Show Connections...</div> </div> <div> <div>Refresh</div> <div>Expand All Connections</div> </div>
<ul style="list-style-type: none"> Client Methods (CMETHOD) Inventory Options (RIMOPTS) Scheduling (TIMER) Virtual Mgr Location (MGRVLOC) 	<div> <div>New Instance</div> <div>Copy Instance</div> <div>Delete Instance</div> <div>Rename Instance</div> <div>Edit Instance...</div> </div> <div> <div>Refresh</div> <div>Expand All Connections</div> </div>

Table A.4 ~ NOVADIGM Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none">Application (ZSERVICE)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Optimize Service...</div><div>Build State File</div></div> <div>Show Connections...</div> <div>Notify Subscribers</div> <div>Prune Below</div> <div>Refresh</div> <div>Expand All Connections</div>
Note: Build State File is available if the Radia Extensions for Windows Installer is installed.	
<ul style="list-style-type: none">Application Packages (PACKAGES)	<div><div>New Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div></div> <div>Copy Package...</div> <div>New Application Wizard...</div> <div>Show Connections...</div> <div>Refresh</div> <div>Expand All Connections</div>
<ul style="list-style-type: none">Desktop (DESKTOP)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div></div> <div>Show Connections...</div> <div>Refresh</div> <div>Expand All Connections</div>

Table A.4 ~ NOVADIGM Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none"> File Resources (FILE) Path (PATH) Unix File Resources (UNIXFILE) 	<div> <div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div> </div>
<ul style="list-style-type: none"> Maintenance Location (MAINTLOC) 	<div> <div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div> </div>

Table A.5 ~ PATCH Instance Context Menu

Instance Name	Context Menu
<ul style="list-style-type: none"> Patch (PATCH) 	<div> <div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div> </div>

Table A.6 ~ POLICY Instance Context Menus	
Instance Name	Context Menu
<ul style="list-style-type: none">Countries (COUNTRY)Mobile Device Config (MBLCONFIG)Multicast (MULTICAST)PDACONFIG (PDACONFIG)Server Stagers (STAGER)	<div><div>New Instance Copy Instance Delete Instance Rename Instance Edit Instance...</div><div>Refresh Expand All Connections</div></div>
<ul style="list-style-type: none">Departments (DEPT)Workgroups (WORKGRP)	<div><div>New Instance Copy Instance Delete Instance Rename Instance Edit Instance...</div><div>Show Connections...</div><div>Refresh Expand All Connections</div></div>
<ul style="list-style-type: none">Users (USER)	<div><div>New Instance Copy Instance Delete Instance Rename Instance Edit Instance... Other Services...</div><div>Show Connections...</div><div>Refresh Expand All Connections</div></div>

Table A.7 ~ SOFTWARE Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none"> Application (ZSERVICE) 	<div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... Optimize Service... </div> <div> Show Connections... </div> <div> Notify Subscribers </div> <div> Refresh Expand All Connections </div>
<ul style="list-style-type: none"> Application Packages (PACKAGE) 	<div> New Instance Delete Instance Rename Instance Edit Instance... </div> <div> Copy Package... New Application Wizard... </div> <div> Show Connections... </div> <div> Refresh Expand All Connections </div>
<ul style="list-style-type: none"> Auto Run (EXECUTE) HTTP Proxy (HTTP) Install Options (INSTALL) Scheduling (TIMER) SD Dependencies (SDDEP) SRV4 Dependencies (SRV4DEP) 	<div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div>

Table A.7 ~ SOFTWARE Instance Context Menus	
Instance Name	Context Menu
<ul style="list-style-type: none">Behavior Services (BEHAVIOR)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Show Connections...</div><div>Refresh</div><div>Expand All Connections</div></div>
<ul style="list-style-type: none">Class Defaults (METACLAS)Desktop (DESKTOP)Dialog Services (DIALOG)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Show Connections...</div><div>Refresh</div><div>Expand All Connections</div></div>
<ul style="list-style-type: none">File Resources (FILE)Mac Alias (MACALIAS)MSI Features (MSIFEATS)MSI Resources (MSI)Panel Service (PANEL)Path (PATH)SD Package (SD)Solaris Package (SVR4)Unix File Resources (UNIXFILE)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Refresh</div><div>Expand All Connections</div></div>
<ul style="list-style-type: none">Registry Resources (REGISTRY)	<div><div>New Instance</div><div>Copy Instance</div><div>Delete Instance</div><div>Rename Instance</div><div>Edit Instance...</div><div>Edit Registry Resource...</div><div>Refresh</div><div>Expand All Connections</div></div>

Table A.7 ~ SOFTWARE Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none"> Version Groups (VGROUP) 	<div> Version Group Editor... </div> <div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div>
<ul style="list-style-type: none"> Versions (VERSION) 	<div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div>

Table A.8 ~ SYSTEM Instance Context Menus

Instance Name	Context Menu
<ul style="list-style-type: none"> Application Manager (ZCOMMAND) 	<div> Execute Command </div> <div> New Instance Copy Instance Delete Instance Rename Instance Edit Instance... </div> <div> Refresh Expand All Connections </div>

Table A.8 ~ SYSTEM Instance Context Menus	
Instance Name	Context Menu
<ul style="list-style-type: none">Consoles (ZCONSOLE)Methods (ZMETHOD)Radia Processes (PROCESS)SQLTABLE (SQLTABLE)System Files (ZFILE)	<div><div>New Instance Copy Instance Delete Instance Rename Instance Edit Instance...</div><div>Refresh Expand All Connections</div></div>

Table A.9 ~ ADMIN Domain Instance Context Menus	
Class Name	Context Menu
<ul style="list-style-type: none">All Class Instances within the ADMIN Domain	<div><div>New Instance Copy Instance Delete Selected Rename Instance Edit Instance...</div></div>

Table A.10 ~ AUDIT Domain Instance Context Menus	
Class Name	Context Menu
<ul style="list-style-type: none">Application (ZSERVICE)	<div><div>New Instance Copy Instance Delete Selected Rename Instance Edit Instance...</div><div>Notify Subscribers</div></div>

Table A.10 ~ AUDIT Domain Instance Context Menus

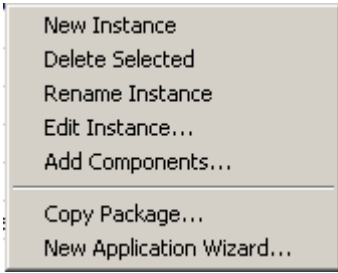
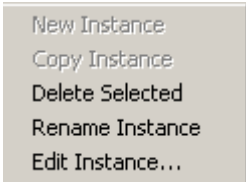
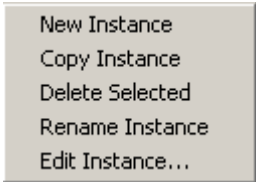
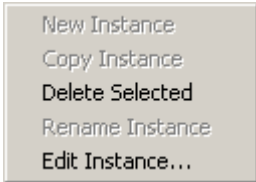
Class Name	Context Menu
<ul style="list-style-type: none"> Audit Packages (PACKAGE) 	
<ul style="list-style-type: none"> Behavior Service (BEHAVIOR) Desktop (DESKTOP) File (FILE) File Scanner (FILESCAN) Path (PATH) WBEM (WBEM) 	
<ul style="list-style-type: none"> Client Methods (CMETHOD) Inventory Options (RIMOPTS) Inventory Scanners (SCANNER) Scheduling (TIMER) Virtual Mgr Location (MGRVLOC) 	
<ul style="list-style-type: none"> File Scanner Filters (FILTER) 	

Table A.11 ~ NOVADIGM Instance Context Menus

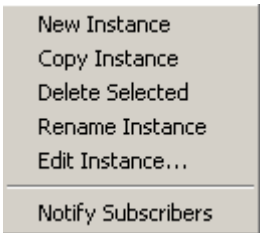
Class Name	Context Menu
<ul style="list-style-type: none"> Application (ZSERVICE) 	

Table A.11 ~ NOVADIGM Instance Context Menus	
Class Name	Context Menu
<ul style="list-style-type: none">Application Packages (PACKAGE)	<div><div>New Instance Delete Selected Rename Instance Edit Instance...</div><div>Copy Package... New Application Wizard...</div></div>
<ul style="list-style-type: none">Desktop (DESKTOP)	<div><div>New Instance Copy Instance Delete Selected Rename Instance Edit Instance...</div></div>
<ul style="list-style-type: none">File Resources (FILE)Path (PATH)Unix File Resources (UNIXFILE)	<div><div>New Instance Copy Instance Delete Selected Rename Instance Edit Instance...</div></div>

Table A.12 ~ PATCH Instance Context Menus	
Class Name	Context Menu
<ul style="list-style-type: none">Patch (PATCH)	<div><div>New Instance Copy Instance Delete Selected Rename Instance Edit Instance...</div></div>

Table A.13 ~ POLICY Instance Context Menus

Class Name	Context Menu
<ul style="list-style-type: none"> All Class Instances within the POLICY Domain 	<div> New Instance Copy Instance Delete Selected Rename Instance Edit Instance... </div>

Table A.14 ~ SOFTWARE Instance Context Menus

Class Name	Context Menu
<ul style="list-style-type: none"> Application (ZSERVICE) 	<div> New Instance Copy Instance Delete Selected Rename Instance Edit Instance... </div> <div> Notify Subscribers </div>
<ul style="list-style-type: none"> Application Packages (PACKAGE) 	<div> New Instance Delete Selected Rename Instance Edit Instance... </div> <div> Copy Package... New Application Wizard... </div>
<ul style="list-style-type: none"> Auto Run (EXECUTE) Class Defaults (METACLAS) Dialog Services (DIALOG) HTTP Proxy (HTTP) Install Options (INSTALL) Scheduling (TIMER) SD Dependencies (SDDEP) SVR4 Dependencies (SVR4DEP) 	<div> New Instance Copy Instance Delete Selected Rename Instance Edit Instance... </div>

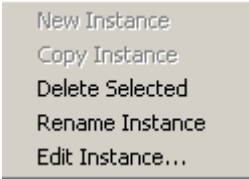
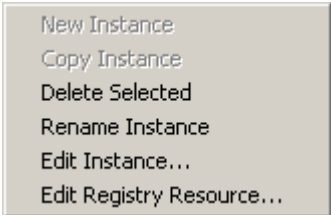
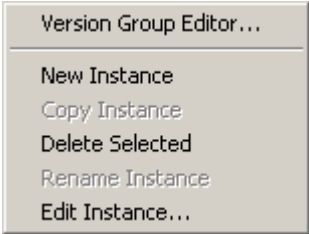
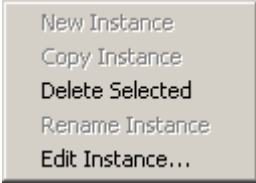
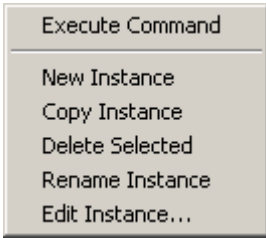
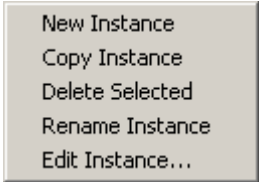
Table A.14 ~ SOFTWARE Instance Context Menus	
Class Name	Context Menu
<ul style="list-style-type: none">• Behavior Services (BEHAVIOR)• Desktop (DESKTOP)• File Resources (FILE)• Mac Alias (MACALIAS)• MSI Features (MSIFEATS)• MSI Resources (MSI)• Panel Services (PANEL)• Path (PATH)• SD Package (SD)• Solaris Package (SVR4)• Unix File Resources (UNIXFILE)	
<ul style="list-style-type: none">• Registry Resources (REGISTRY)	
<ul style="list-style-type: none">• Version Groups (VGROUP)	
<ul style="list-style-type: none">• Versions (VERSION)	

Table A.15 ~ SYSTEM Instance Context Menus

Class Name	Context Menu
<ul style="list-style-type: none"> Application Manager (ZCOMMAND) 	
<ul style="list-style-type: none"> Consoles (CONSOLE) Methods (ZMETHOD) Radia Processes (PROCESS) SQLTABLE (SQLTABLE) System Files (ZFILE) 	



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 System Explorer.
- Be able to replace the entire contents of the data distributed for a file within a published package in the Radia System Explorer.
- Be able to add a component instance for a file or behavior to a software package, without having to republish the package.

Urgent Warning!

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 clients in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Radia would detect a package change and automatically distribute the change to all authorized clients.

The shortcut menu commands for component modifications are shown in *Figure B.1 ~ Instance shortcut menu showing Edit, Replace, and Add Component commands* below.

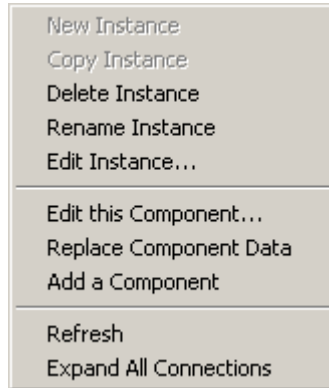


Figure B.1 ~ Instance shortcut menu showing Edit, Replace, and Add Component commands.

When to Use these Commands

Examples of when you may want to use this option to modify component instances are listed *Table B.1 ~ When to Use the Advanced Edit Component Commands* below.

Table B.1 ~ When to Use the Advanced Edit Component Commands	
Command	Typical Use
Edit this Component	<ul style="list-style-type: none">• Change a serial number deployed with a file.• Edit the contents of a readme file, document, spread sheet, logo image, or PDF.
Replace Component Data	<ul style="list-style-type: none">• Replace the entire contents of an existing file in a package with contents of another file (outside of Radia Database).
Add a Component	<ul style="list-style-type: none">• 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**.

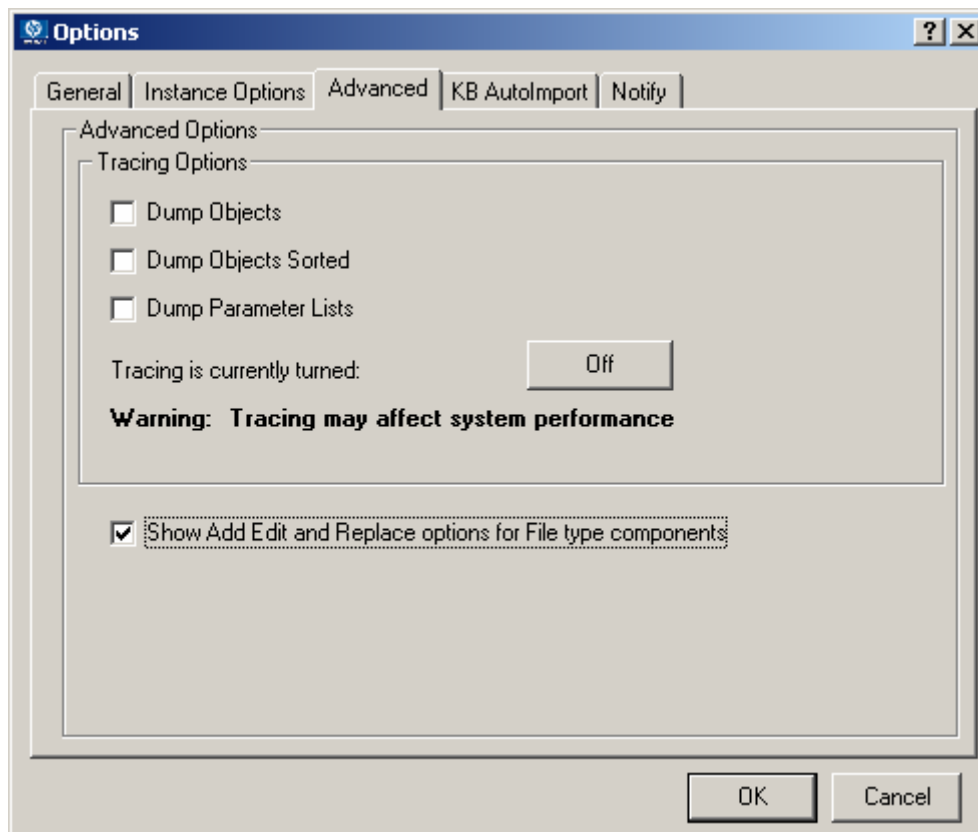


Figure B.2 ~ Options window, Advanced tab includes Show Add Edit and Replace options for File type components.

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.

Urgent Warning!

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 clients in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Radia would detect a package change and automatically distribute the change to all authorized clients.

Selecting this option displays the following component modification commands on the shortcut menus for component instances.

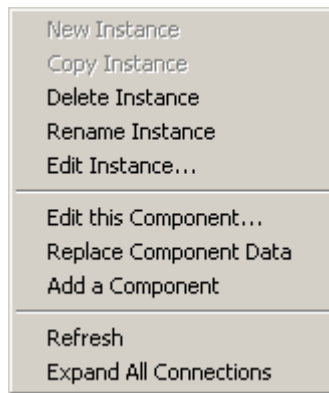


Figure B.3 ~ Instance shortcut menu showing Edit, Replace, and Add Component commands.

Editing a Component

The **Edit this Component** command 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 Radia Configuration Server and the instance in the Radia Database is updated.

Urgent Warning!

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 clients in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Radia would detect a package change and automatically distribute the change to all authorized clients.

For example, you can edit the distributable contents of any component instance that has been packaged and is represented in the Radia System Explorer. 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 Radia System Explorer takes care of the publishing for you, saving you from having to republish the package.

The original component's contents are not retained within the Radia System Explorer.

See the *Radia Application Manager Guide* for information on creating and promoting packages.

To edit a component

1. In the Radia System Explorer, 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 Radia Database.
3. Right-click the instance in the tree view. A context menu opens.

*Figure B.4 ~ Right-click the file instance to be modified to display the shortcut menu on page 288 shows an example of the context menu for the Amortize.TXT readme file instance within the Amortize NT/2000/XP package. It includes the commands: **Edit this Component**, **Replace Component Data**, and **Add a Component**.*

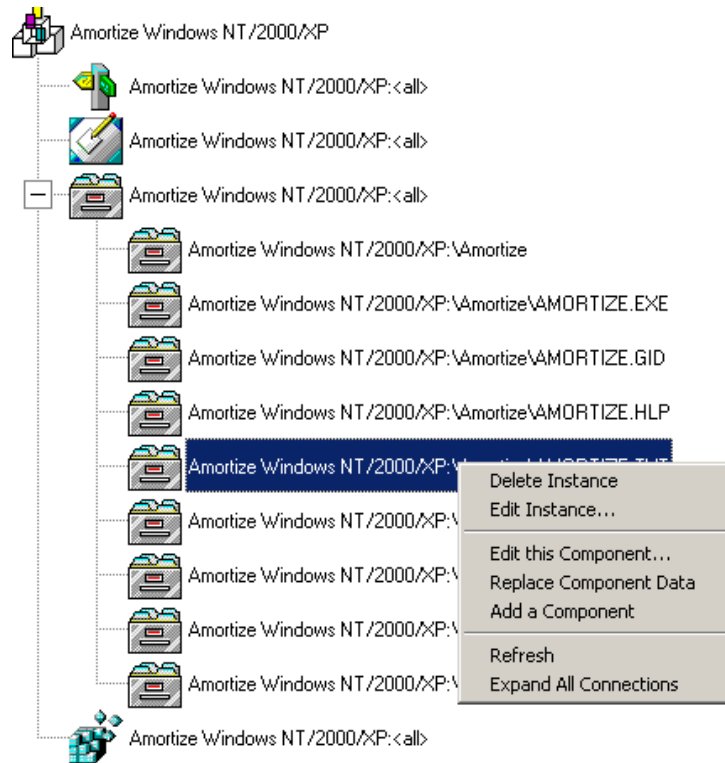


Figure B.4 ~ Right-click the file instance to be modified to display the shortcut menu.

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.

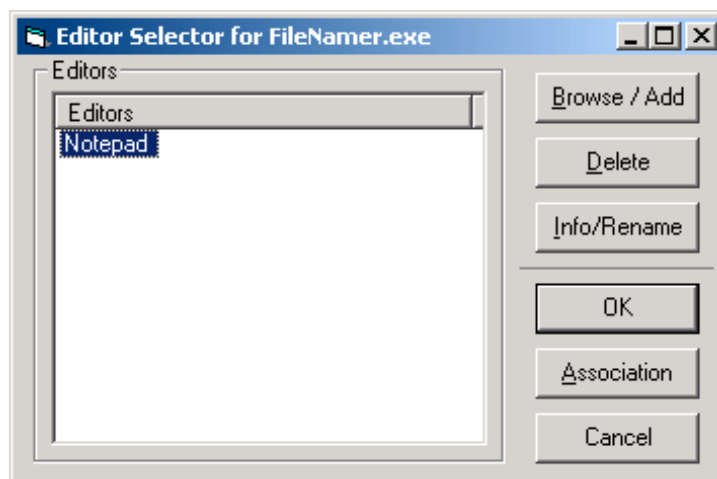


Figure B.5 ~ The Editor Selector dialog box.

For details on using this dialog box, see *Using the Editor Selector Dialog Box* on page 291.

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*.

Caution

If you save the file with a different name, the Radia component editor cannot make the changes to the file you selected from the Radia System Explorer.

7. Close the editor.
8. If necessary, close the **Editor Selector** dialog box using **OK** to save the changes to the instance.

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 have Radia 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.

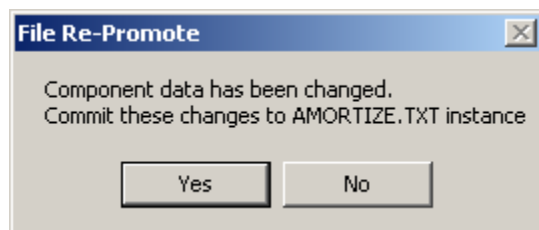


Figure B.6 ~ Use the File Re-Promote dialog box to commit your edits to the component data.

After you commit the changes, the instance attributes for ZRSCDATE, ZRSC TIME, and ZRSC SIZE 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...
ZRSCMO	Mandatory/Optional on Client [M/O]	0
ZRSCVRFY	Verify Resource File on Connect	Y
ZRSCCONF	Confirm File Download [Y/N]	Y
ZRSCCRASH	DOS File Attribute [R/A/S/H]	A
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY
ZRSCMMEM	PDS Member Name	21859BAE
ZRSCDATE	Resource Date Stamp - From Promote	20021023
ZRSC TIME	Resource Time Stamp - From Promote	16:27:40
ZRSC SIZE	Resource Size - From Promote	00000001193
ZRSCVERS	Resource Version - From Promote	
ZRSCCSTA	Client File Status	999
ZRSCCKPT	Resource Checkpoint	
ZRSCCRC	Resource CRC	

Figure B.7 ~ File's instance attributes for date, time, and size reflect changes from the edit session.

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.

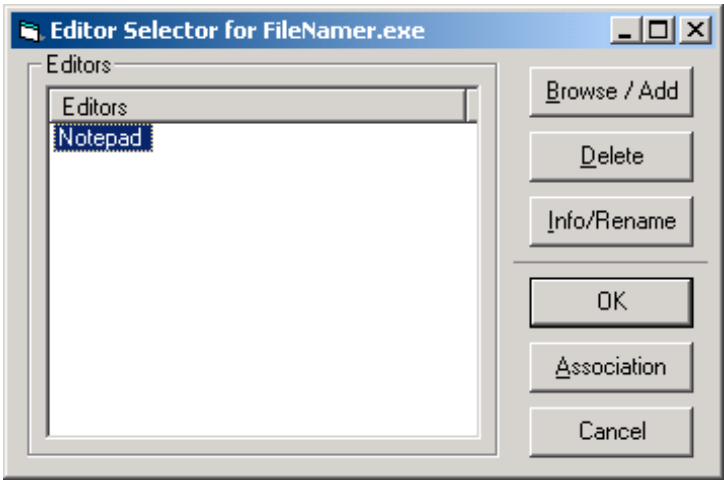


Figure B.8 ~ Editor Selector dialog box to Edit a Component.

Table B.2 ~ Editor Selector Dialog Box Buttons below summarizes the **Editor Selector** dialog box button actions.

Table B.2 ~ Editor Selector Dialog Box Buttons	
Button	Action
Browse/Add	Opens a File Browse dialog box to 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.

Table B.2 ~ Editor Selector Dialog Box Buttons

Button	Action
Info/Rename	For the currently 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 currently 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	Cancels the Edit a Component command without making any changes to the selected component or package.

Replacing Component Data

Use the **Replace Component Data** command when you need to completely replace the entire contents of a packaged file with the contents of a file that exists outside of the database. For example, suppose there's a new Readme.TXT, and you want to replace its entire contents. Prepare a revised Readme.TXT file outside of the Radia System Explorer. Then, use the **Replace Component Data** command from within the Radia System Explorer 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 will change, just its distributed contents.

Urgent Warning!

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 clients in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Radia would detect a package change and automatically distribute the change to all authorized clients.

To replace [all] component data

1. Prepare a file outside of Radia System Explorer whose contents represent the replacement data for a packaged component file in the Radia Database. The prepared file can have any name.

Caution

The entire contents of the prepared file will replace the entire data contents of the file selected from the Radia System Explorer using the **Replace Component Data** command. Verify the contents of the files prior to continuing with this procedure.

2. In the Radia System Explorer, 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 Radia Database.
4. Right-click the instance in the tree view. A context menu opens.

The following figure shows an example of the context menu for the Amortize.TXT readme file instance within the Amortize NT/2000/XP package. It includes the commands: **Edit this Component**, **Replace Component Data**, and **Add a Component**.

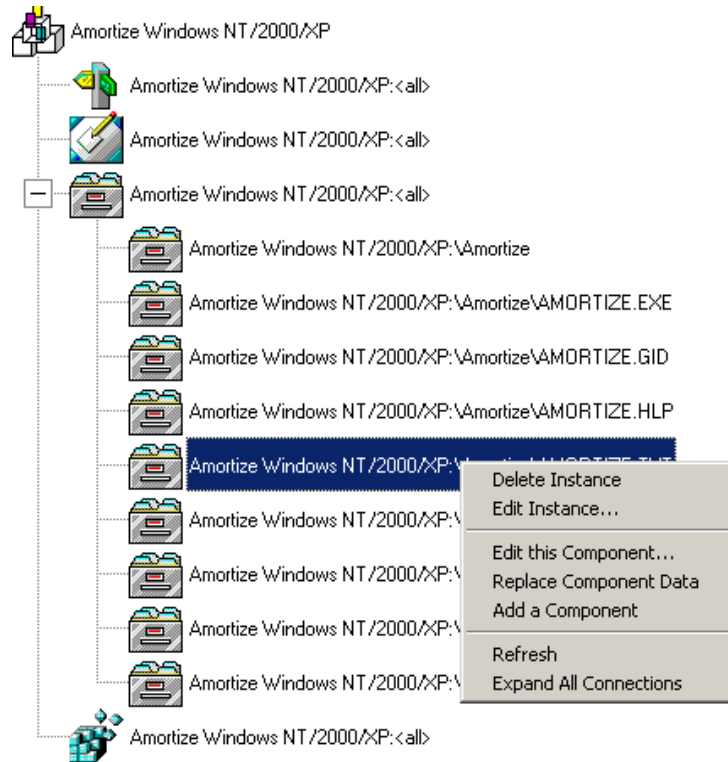


Figure B.9 ~ Right-click the file instance whose data is to be replaced and select *Replace Component Data* from the shortcut menu.

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 Radia System Explorer.

The following figure shows an example of navigating to a prepared file named AMORTIZE2.TXT.

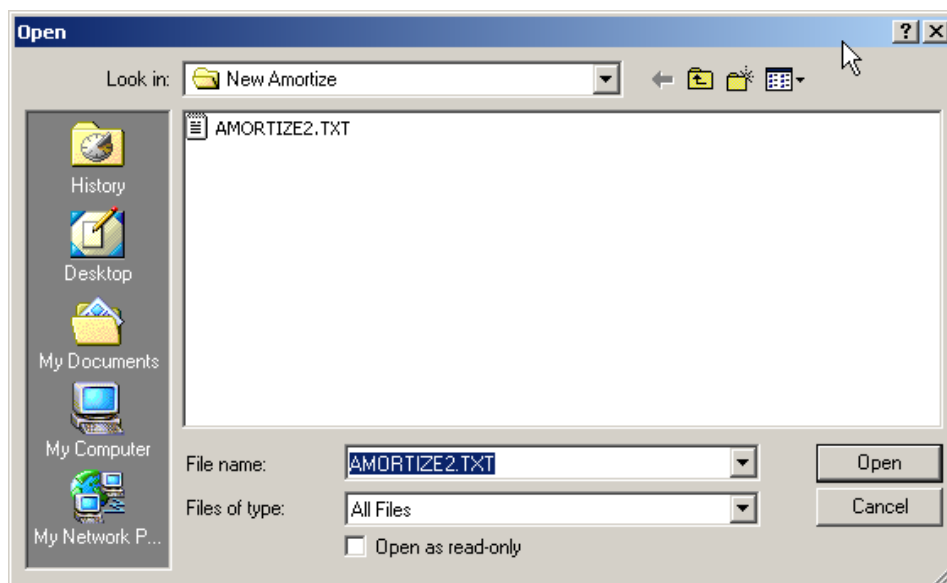


Figure B.10 ~ Navigate to the file whose contents will replace the data in AMORTIZE.TXT.

7. Click **Open** to select the file with the replacement data.

The following dialog box opens to confirm the data replacement.

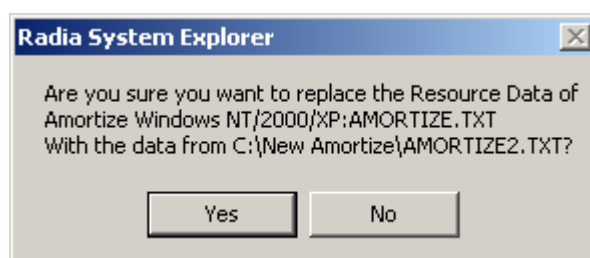


Figure B.11 ~ Confirmation dialog box for replacing data in a component.

8. Click **Yes** to replace the data, click **No** to abort the data replacement.
 - If you click **Yes**, the Radia System Explorer 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 in *Figure B.7 ~ File's instance attributes for date, time, and size reflect changes from the edit session* on page 290.

- If you click **No**, the replacement command is cancelled.

Adding a Component

The **Add a Component** command is only available for file and behavior components. 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 Radia Packager.

Urgent Warning!

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 clients in a production environment. Using this feature in such an environment would likely result in massive distributions of unnecessary package updates, since Radia would detect a package change and automatically distribute the change to all authorized clients.

New files are added by selecting an instance to use as a template. The Radia System Explorer will derive the path for the new instance from the template's LOCATION variable. Once a template is selected, right-click it to display the context 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 Radia System Explorer, 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 context menu opens.

The following figure shows an example of the context menu for the Amortize.TXT readme file instance within the Amortize NT/2000/XP package. It includes the commands: **Edit this Component**, **Replace Component Data**, and **Add a Component**.

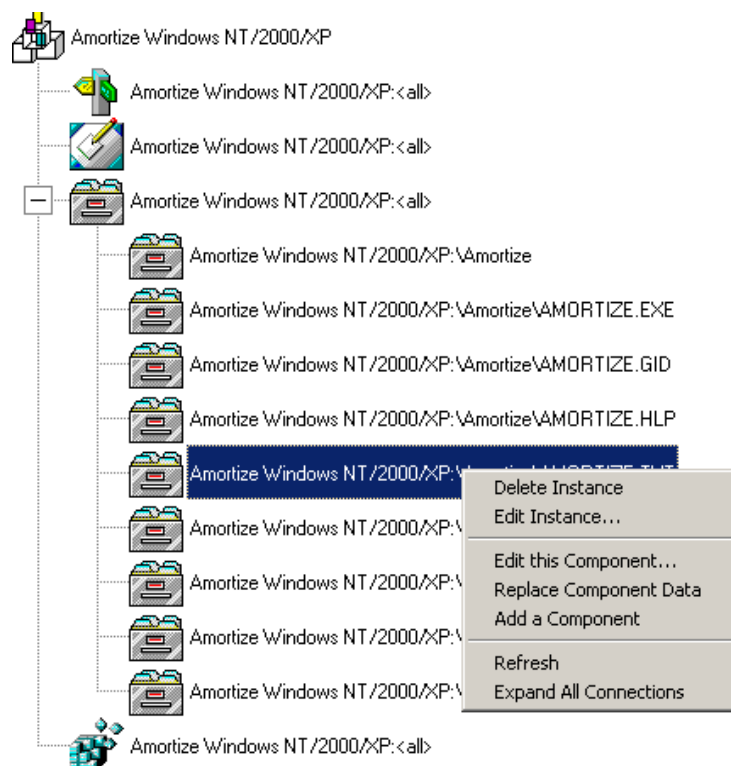


Figure B.12 ~ Right-click the file instance whose location can serve as a template for the added component and select Add a Component from the shortcut menu.

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.

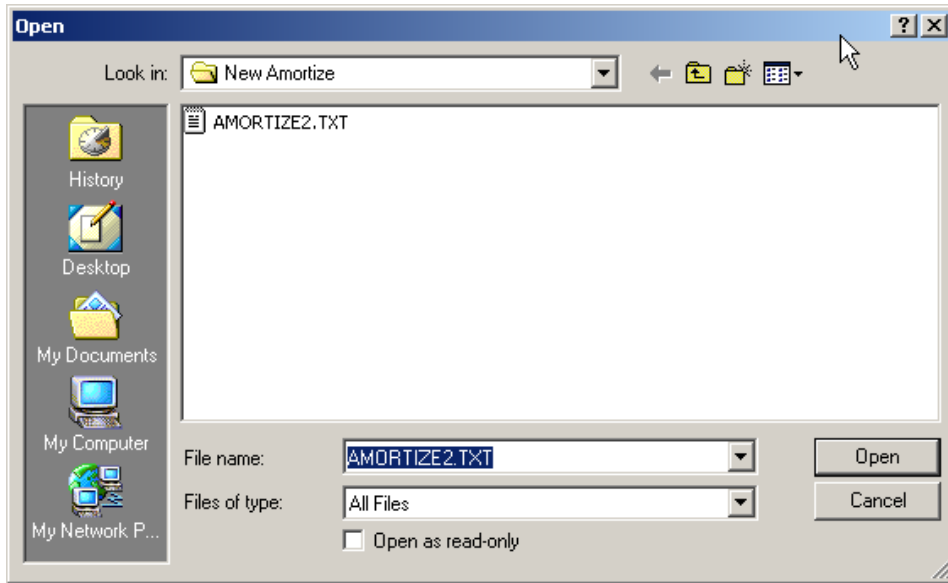


Figure B.13 ~ Navigate to the file to be added to the package.

6. Click **Open** to select the file to be added with the same LOCATION as the template file. The following dialog box opens to confirm the addition.
-

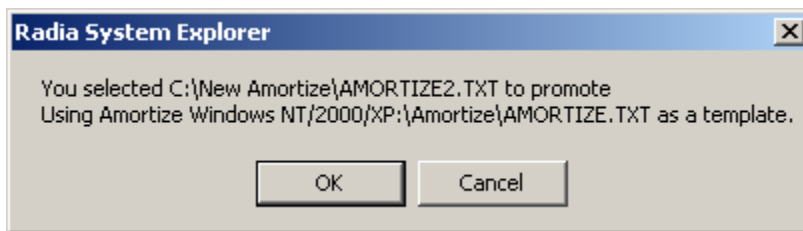


Figure B.14 ~ Confirmation dialog box when adding a component instance.

7. Click **OK** or **Cancel**.
- If you click **OK**, the Radia System Explorer 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.

Summary

- Use the Advanced Options that allow you to Add, Edit, and Replace File-type components in *non-production environments* only.
- Use **Edit this Component** to dynamically modify the distributed contents of a component from within the Radia System Explorer, without having to republish the entire package.
- Use **Replace Component Data** to replace all of the distributed data in a component instance with the contents of a prepared file outside of the Radia Database. You don't have to republish the package using this command.
- Use the **Add a Component** command to add an instance to a package without having to promote the file and re-publish the package. An existing instance must be able to serve as a template for the added component's path LOCATION.

Figures

Figure 2.1 ~ Opening the Radia System Explorer.....	23
Figure 2.2 ~ The Radia System Explorer Security Information dialog box.	23
Figure 2.3 ~ Radia System Explorer window.	24
Figure 2.4 ~ You can manipulate the view of the window by clicking and dragging borders.....	25
Figure 2.5 ~ The File menu.....	26
Figure 2.6 ~ The Edit menu.....	26
Figure 2.7 ~ The View menu.....	27
Figure 2.8 ~ The Window menu.	28
Figure 2.9 ~ The Help menu.	29
Figure 2.10 ~ The Radia System Explorer toolbar.	31
Figure 2.11 ~ The Options dialog box of Radia System Explorer with the General tab displayed.	32
Figure 2.12 ~ The Instance Options tab of the Options dialog box.	35
Figure 2.13 ~ The Advanced tab of the Options dialog box.....	37
Figure 2.14 ~ The KB AutoImport tab of the Options dialog box.....	39
Figure 2.15 ~ The Notify tab of the Options dialog box	41
Figure 3.1 ~ The Radia Database□LICENSE and PRIMARY files.....	44
Figure 3.2 ~ Navigating through the tree structure.	46
Figure 3.3 ~ Clicking the minus sign closes the tree view below that level.....	47

Figure 3.4 ~ Use the Prune Below function to discard the history of previously opened branches beneath a component.....	48
Figure 3.5 ~ Use the Refresh function to rebuild the display of a node and its branches in the tree view.	49
Figure 3.6 ~ Compound names are used to manage large numbers of instances.....	51
Figure 3.7 ~ An example of a BASE_INSTANCE in the Application class.	52
Figure 3.8 ~ Create Domain dialog box.	54
Figure 3.9 ~ Copy Class dialog box.	55
Figure 3.10 ~ The Class Editing dialog box opens when the option Open Class Editor When a Class is Copied is active.	56
Figure 3.11 ~ Class Edit Confirmation dialog box.	57
Figure 3.12 ~ Create Class dialog box.	58
Figure 3.13 ~ Editing Class dialog box.	59
Figure 3.14 ~ Class Edit Confirmation dialog box.	60
Figure 3.15 ~ The Editing Class dialog box.	61
Figure 3.16 ~ The Attribute Description for the ZSVCONF variable terminates with [Y/N], indicating a flag attribute.	68
Figure 3.17 ~ An attribute terminating with one-character choices separated by slashes and enclosed in brackets is treated as a flag set.....	69
Figure 3.18 ~ An attribute description terminating with a series of choices separated by slashes and enclosed in brackets is offered a drop-down list for editing attributes.	70
Figure 3.19 ~ Confirm Restore dialog box.	70
Figure 3.20 ~ Warning dialog box for blank attribute fields.	71
Figure 3.21 ~ Class Edit Confirmation dialog box.	71
Figure 3.22 ~ Confirm Delete dialog box for deleting a class.	72
Figure 3.23 ~ Copy Instance dialog box.	73
Figure 3.24 ~ Create Instance dialog box.	74
Figure 3.25 ~ Editing Instance dialog box.	75
Figure 3.26 ~ Instance Edit Confirmation dialog box.	76
Figure 3.27 ~ Initiating a multiple instance edit.	77

Figure 3.28 ~ Select Edit Multiple Instances.	78
Figure 3.29 ~ Editing Multiple Instances dialog box.	78
Figure 3.30 ~ The Change All button has been changed to Reset.	79
Figure 3.31 ~ The Change All button is active, indicating no change was made to this highlighted variable during the current Editing Instance dialog box session.	80
Figure 3.32 ~ The Confirm Restore message.	80
Figure 3.33 ~ Multiple Instance Edit Confirmation dialog box.	81
Figure 3.34 ~ Rename Instance dialog box.	82
Figure 3.35 ~ Confirm Delete dialog box.	83
Figure 4.1 ~ PRIMARY.POLICY.Users class.	86
Figure 4.2 ~ Create a Policy for your organization.	88
Figure 4.3 ~ To create a new user, right-click the Users class. Select New Instance from the shortcut menu.	89
Figure 4.4 ~ Create Instance dialog box.	90
Figure 4.5 ~ Completed Create Instance dialog box.	90
Figure 4.6 ~ A new user, esb1, has been added to the Users class.	91
Figure 4.7 ~ Application Packages (PACKAGE) class	92
Figure 4.8 ~ Sample Application (ZSERVICE) instances as shipped from HP.	93
Figure 4.9 ~ An example of the Drag & View service and its connections.	94
Figure 4.10 ~ Each variable can hold a connection to another package.	95
Figure 4.11 ~ Select New Instance from the shortcut menu.	98
Figure 4.12 ~ Create Instance dialog box.	99
Figure 4.13 ~ The newly created Sales Department in the Departments (DEPT) class.	99
Figure 4.14 ~ The circle with a slash indicates that a connection is not allowed.	101
Figure 4.15 ~ The paperclip indicates that a connection is allowed.	101
Figure 4.16 ~ The Select Connection Attribute dialog box showing the next possible connection.	102
Figure 4.17 ~ Select Show Connections to open the Show connectable classes dialog box.	103
Figure 4.18 ~ Show connectable classes dialog box.	103
Figure 4.19 ~ Domain list for connectable classes.	104

Figure 4.20 ~ Warning message for a non-permitted connection.....	104
Figure 4.21 ~ Expanded PRIMARY.POLICY.Users (USER) class.....	106
Figure 4.22 ~ Select Show Connections from the shortcut menu.....	107
Figure 4.23 ~ Show connectable classes for domain dialog box.....	108
Figure 4.24 ~ All of the Departments (DEPT) class instances are now displayed.....	109
Figure 4.25 ~ Instance Connection confirmation box.....	110
Figure 4.26 ~ Select Connection Attribute dialog box that opens after dragging a FILE instance onto a PACKAGE instance.	111
Figure 4.27 ~ Select SOFTWARE from the drop-down list.	113
Figure 4.28 ~ The ZSERVICE in the list view.....	114
Figure 4.29 ~ Select Amortize and drag-and-drop it onto the Sales Departments (DEPT) instance.....	115
Figure 4.30 ~ All users within the Sales Department (DEPT) will receive the Amortize Service.....	116
Figure 5.1 ~ The GS-CALC PACKAGE class instance.	121
Figure 5.2 ~ Select New Application Wizard from the shortcut menu	122
Figure 5.3 ~ New Application Wizard dialog box.....	123
Figure 5.4 ~ Application Target Type dialog box.	126
Figure 5.5 ~ Application Properties dialog box with application properties entered in the appropriate fields.	127
Figure 5.6 ~ Application Level Event Reporting dialog box for reporting application level events to the Radia Configuration Server.....	129
Figure 5.7 ~ The Application Summary dialog box summarizes and confirms selections made in the New Application Wizard.....	130
Figure 5.8 ~ Application successfully added confirmation box.....	131
Figure 5.9 ~ The highlighted GS-Calc package is the service you just created.....	132
Figure 5.10 ~ New Application Wizard with MSI Basic Behaviors section.....	133
Figure 5.11 ~ Create Instance dialog box.	135
Figure 5.12 ~ Editing Version Group dialog box.	136
Figure 5.13 ~ The Editing Version Group dialog box displays time and date controls when the Activate On or After check box is selected.....	137
Figure 5.14 ~ Create Version dialog box.....	138

Figure 5.15 ~ Version Editor dialog box.....	139
Figure 5.16 ~ Connect the Amortize Windows 95/98 package to the Hello World Version Group.....	140
Figure 5.17 ~ A Version instance created and listed in the Unassigned Versions list.....	141
Figure 5.18 ~ Editing Version Group dialog box showing Version1 assigned to Hello World.	142
Figure 5.19 ~ The selected version for deployment, Version 2, is highlighted in the Version Group Layout list.	143
Figure 5.20 ~ Version Group class shortcut menu.	144
Figure 5.21 ~ PATCH.CLASS in the PRIMARY file.	146
Figure 5.22 ~ BLDTEST packages in the SOFTWARE domain.....	149
Figure 5.23 ~ ZSERVICE shortcut menu.....	150
Figure 5.24 ~ BLD_CREATE ZSERVICE instance.....	150
Figure 5.25 ~ BLD_CREATE ZSERVICE instances.	151
Figure 5.26 ~ Optimize Service from the BLD_CREATE ZSERVICE.	152
Figure 5.27 ~ Service Optimization Wizard—Available Packages dialog box.....	153
Figure 5.28 ~ Select which package you would like to optimize.....	154
Figure 5.29 ~ Select the package you would like to compare the " from" against.	155
Figure 5.30 ~ Completed Service Optimization Wizard.....	156
Figure 5.31 ~Files and Progress dialog box.	157
Figure 5.32 ~ Enabled buttons for patch building.....	159
Figure 5.33 ~ Auto Refresh Timer dialog box—Enter an interval, in minutes, to automatically refresh the information in the Status column.....	160
Figure 5.34 ~ Service Optimization—Selecting the files to build a patch.....	161
Figure 5.35 ~ Build confirmation dialog box.....	161
Figure 5.36 ~ Patch in the PATCH class of the PATCH domain.....	163
Figure 5.37 ~ The MESSAGE instance in the patch.....	164
Figure 5.38 ~ ZSERVICE sample services.....	167
Figure 5.39 ~ Creating a new ZSERVICE instance.	168
Figure 5.40 ~ Create a new instance – Sales Service.....	169
Figure 5.41 ~ Newly created instance—Sales Services.....	169

Figure 5.42 ~ Select Show Connections from the shortcut menu.....	170
Figure 5.43 ~ SOFTWARE.ZSERVICE Connections dialog box.	171
Figure 5.44 ~ The Application (ZSERVICE) Class instances are displayed in the list view.	172
Figure 5.45 ~ The completed Master Group—Sales Services.	173
Figure 5.46 ~ Menu based notify—shortcut menu.	175
Figure 5.47 ~ Notify Start dialog box.	176
Figure 5.48 ~ Audience List Error.	176
Figure 5.49 ~ Radia Notify Manager dialog box.	177
Figure 5.50 ~ Radia Notify Manager allows three options for notification.	178
Figure 5.51 ~ Selecting the Send an Email option will display a view similar to this dialog box.	179
Figure 5.52 ~ After selecting the notification type, a Notification Details dialog box opens.	180
Figure 5.53 ~ Check the Use Custom Notify Domain box to specify a domain name.	181
Figure 5.54 ~ Selecting the Deferred Notify box will display the time and date controls.	182
Figure 5.55 ~ After selecting the notification type and details, a summary screen opens.	183
Figure 5.56 ~ Notify Initiated dialog box.	184
Figure 5.57 ~ The Notify Status dialog box displays the status of the notification.	184
Figure 5.58 ~ After an initial notification, the Notify file is listed among the files of the Database in the Radia System Explorer tree view.	185
Figure 5.59 ~ Notify file—shortcut menu.	185
Figure 5.60 ~ Excerpt from the Radia Configuration Server EDMPROF.DAT file.	187
Figure 5.61 ~ Class shortcut menu.	189
Figure 5.62 ~ Filter Instances dialog box.	190
Figure 5.63 ~ Filter specification for the Users class is displayed under Current Settings.	190
Figure 5.64 ~ Filter applied to the Users class.	191
Figure 5.65 ~ Filtering component classes like the File class produces this dialog box.	193
Figure 5.66 ~ This figure shows all of the FILE class instances that were filtered using the Owning Package list.	194
Figure 5.67 ~ The RAD_MAST instance has a connection to an instance of the ZADMIN class as displayed in the list view of the Radia System Explorer window.	195

Figure 5.68 ~ The value of the ZADMFUNC variable is set to MASKINST to retrieve a list of instance masks.....	196
Figure 5.69 ~ A connection is further made to ZLIST32.MASTER_SYSTEM_INSTANCES.....	197
Figure 6.1 ~ The list view in the Radia Client Explorer displays the instance name, the number of heaps or instances, the size of the instance, and the last date and time the instance was modified.....	201
Figure 6.2 ~ Double-click on an instance to display the objects inside.	202
Figure 6.3 ~ Location of the ZMASTER object in the Radia Database.	204
Figure 6.4 ~ The Radia Configuration Server performs resolution.....	205
Figure 6.5 ~ The PRIMARY.SYSTEM.PROCESS.ZMASTER instance.....	206
Figure 6.6 ~ POLICY.USER.ALEE instance.	207
Figure 6.7 ~ The ZSERVICE.AMORTIZE instance.	208
Figure 6.8 ~ The ZMETHOD.PUTPROF_ZMASTER instance.	209
Figure 7.1 ~ The MASTER_CONNECTION_RULES instance of the PRIMARY file ADMIN domain Connection Rules (ZCONNECT) class is highlighted.	212
Figure 7.2 ~ The Name Lists (ZLIST) class in the ADMIN domain.....	213
Figure 7.3 ~ The ZLIST.CONNECT_USER_TO_RULES instance.....	214
Figure 7.4 ~ The connectable classes for the POLICY domain.....	215
Figure 7.5 ~ The CONNECT_TEST_TO_RULES in the ZLIST class.....	217
Figure 7.6 ~ The connectable classes for our TEST class.	218
Figure 7.7 ~ Add the new class's name to the list of variables in the Connection Rules class.....	219
Figure 7.8 ~ Set the variable in the connection rule.	220
Figure 7.9 ~ The completed connection rule for our TEST class in the MASTER_CONNECTION_RULES.....	221
Figure 8.1 ~ Accessing the Feature Set Editor thru the Radia System Explorer tree view.	225
Figure 8.2 ~ The Feature Set Editor dialog box.	226
Figure 8.3 ~ Child features displayed.	227
Figure 8.4 ~ Child features closed under Microsoft Word.....	228
Figure 8.5 ~ The Features menu opens when the drop-down arrow next to a feature is clicked.	229
Figure 8.6 ~ The graphic settings for features.	233

Figure 8.7 ~ Names of features appearing in boldface indicate a change to their attributes has occurred.....	234
Figure 8.8 ~ A shortcut menu for Hide and Disable opens when a feature's name is right-clicked.....	235
Figure 8.9 ~ All hidden features appear with the prefix (H) before their names.....	236
Figure 8.10 ~ All disabled features appear with the prefix (D) before their names.....	237
Figure 8.11 ~ The Editing DATA1.MSI Instance dialog box is used to change the value of SHOWINCP to N.....	239
Figure 8.12 ~ Copying a package.	241
Figure 8.13 ~ Copy Package dialog box.	242
Figure 8.14 ~ Copy Package and Components dialog box.	242
Figure 8.15 ~ Opening the Feature Set Editor from the shortcut menu.	243
Figure 8.16 ~ The Feature Set Editor displaying applications whose properties are set to Ignored.....	244
Figure 9.1 ~ Accessing Registry File Editor.	251
Figure 9.2 ~ Select Edit Registry Resource from the context menu.....	251
Figure 9.3 ~ Registry File Editor.	252
Figure 9.4 ~ Expand a node.....	253
Figure 9.5 ~ Collapse a node.	254
Figure 9.6 ~ Setting node properties.....	254
Figure 9.7 ~ Select the Registry Instance of the package.....	257
Figure 9.8 ~ Select the key you are editing by double-clicking on the value name in the list view.....	258
Figure 9.9 ~ Override Value dialog box.	259
Figure 9.10 ~ Exiting the Registry dialog box.	260
Figure B.1 ~ Instance shortcut menu showing Edit, Replace, and Add Component commands... ..	284
Figure B.2 ~ Options window, Advanced tab includes Show Add Edit and Replace options for File type components.	285
Figure B.3 ~ Instance shortcut menu showing Edit, Replace, and Add Component commands... ..	286
Figure B.4 ~ Right-click the file instance to be modified to display the shortcut menu.	288
Figure B.5 ~ The Editor Selector dialog box.....	289
Figure B.6 ~ Use the File Re-Promote dialog box to commit your edits to the component data. ..	290

Figure B.7 ~ File's instance attributes for date, time, and size reflect changes from the edit session.	290
Figure B.8 ~ Editor Selector dialog box to Edit a Component.	291
Figure B.9 ~ Right-click the file instance whose data is to be replaced and select Replace Component Data from the shortcut menu.	294
Figure B.10 ~ Navigate to the file whose contents will replace the data in AMORTIZE.TXT.	295
Figure B.11 ~ Confirmation dialog box for replacing data in a component.	295
Figure B.12 ~ Right-click the file instance whose location can serve as a template for the added component and select Add a Component from the shortcut menu.	297
Figure B.13 ~ Navigate to the file to be added to the package.	298
Figure B.14 ~ Confirmation dialog box when adding a component instance.	298

Tables

Table P.1 ~ Styles.....	7
Table P.2 ~ Usage.....	7
Table P.3 ~ Terminology*	8
Table 1.1 ~ Key Radia Terms	18
Table 2.1 ~ Summary of Mouse Actions.....	22
Table 2.2 ~ Edit Menu	26
Table 2.3 ~ View Menu	27
Table 2.4 ~ Window Menu	28
Table 2.5 ~ Help Menu.....	29
Table 2.6 ~ Radia System Explorer Toolbar	31
Table 2.7 ~ Settings on the General Tab and their Functions	34
Table 2.8 ~ Settings on the Instance Options Tab	36
Table 2.9 ~ Advanced Options Tab.....	37
Table 2.10 ~ Settings on the Notify Options Tab	41
Table 3.1 ~ Class Information Properties	62
Table 3.2 ~ Class Attribute Types and their Functions.....	63
Table 3.3 ~ Manager Properties and their Functions	65
Table 4.1 ~ Policy Domain	87
Table 5.1 ~ Variables Populated by the New Application Wizard	125
Table 5.2 ~ Application Properties Data Entry Fields.....	127
Table 5.3 ~ Event Reporting Keywords and their Denotations.....	131
Table 5.4 ~ Values Associated with each Event Reporting Keyword.....	131
Table 5.5 ~ Attributes in the Patch Class.....	146
Table 5.6 ~ Files and Progress Information	158
Table 5.7 ~ Patch Criteria Information	159
Table 5.8 ~ Button Functions	159

Table 5.9 ~ ZSVCGRP Variable	166
Table 5.10 ~ Examples of Valid Filter Expressions	188
Table 8.1 ~ Features Menu Options	230
Table 8.2 ~ Feature Set Editor Buttons	238
Table 9.1 ~ Registry File Editor Context Menu Options	255
Table A.1 ~ Primary, Domain and Class Shortcut Menus	268
Table A.2 ~ ADMIN Instance Context Menus	269
Table A.3 ~ AUDIT Instance Context Menus	269
Table A.4 ~ NOVADIGM Instance Context Menus	271
Table A.5 ~ PATCH Instance Context Menu	272
Table A.6 ~ POLICY Instance Context Menus	273
Table A.7 ~ SOFTWARE Instance Context Menus	274
Table A.8 ~ SYSTEM Instance Context Menus	276
Table A.9 ~ ADMIN Domain Instance Context Menus	277
Table A.10 ~ AUDIT Domain Instance Context Menus	277
Table A.11 ~ NOVADIGM Instance Context Menus	278
Table A.12 ~ PATCH Instance Context Menus	279
Table A.13 ~ POLICY Instance Context Menus	280
Table A.14 ~ SOFTWARE Instance Context Menus	280
Table A.15 ~ SYSTEM Instance Context Menus	282
Table B.1 ~ When to Use the Advanced Edit Component Commands	284
Table B.2 ~ Editor Selector Dialog Box Buttons	291

Procedures

To open the Radia System Explorer.....	22
To specify the Knowledge Base Manager AutoImport options.....	39
To navigate through the tree structure	46
To use the Prune Below function	48
To rebuild the display of a node and its branches in the tree view	49
To add a domain to a file.....	54
To copy an existing class.....	55
To add a new class from scratch.....	57
To edit a class	61
To delete a class and all of its instances	72
To copy an existing instance.....	73
To add a new instance from scratch.....	74
To edit an instance	75
To initiate and execute a multiple instance edit	77
To undo all changes you've made in the current Editing Instance dialog box session.....	80
To rename an instance.....	82
To delete an instance	83
To create a new user	88
To create a Departments (DEPT) instance.....	97
To connect a User to a Departments (DEPT) instance	105
To connect a service to a policy.....	112
To use the New Application Wizard to create a GS-CALC service	120
To define the MSI Basic Behaviors section of the New Application Wizard	133
To create a Version Group instance for a package	134
To assign a Version Instance to a Version Group	141
To remove a Version Assignment in a Version Group	144

To delete a Version Instance from a Version Group.....	144
To edit a Version Group	144
To configure the Optimize Service feature.....	149
To create a Master Group	167
To create a notification event.....	175
To specify a domain name	181
To defer notification until a specific date and time	182
To set a permanent filter to filter instances of the USER class on the prefix ESB_	188
To remove a permanent filter	191
To set a temporary filter to filter instances of the USER class on the prefix ESB_.....	192
To remove a temporary filter	192
To view the base or null instance	192
To filter FILE instances to select only those that are connected to a particular PACKAGE instance	194
To access the Feature Set Editor	225
To make a new variant set of features	241
To access the Registry File Editor	250
To expand a single node in the tree view	253
To collapse a single node on the tree	253
To edit the data in a registry key	256
To restore the data in a registry key	259
To close the Registry File Editor	260
To edit a component	287
To replace [all] component data.....	292
To add a component to a package	296

Index

A

A Class is Copied check box	34
A New Class is Added check box	34
Add a Component	
description	296
when to use	284
Add/Remove Programs applet.....	239
ADMIN domain.....	45
Administrative Installation Point	230, 241
alias name	67
always connection	64
application instance, creating	120
Application Management client.....	264
Association button	292
attribute	
alias	67
definition	18
Attribute Contains the Instance Alias Name	67
attribute type	
connection	64
expression	63
method	64
variable	63
attribute value, description.....	18

B

base instance	
description	52
inheritance	52
Browse/Add button	291
byte level differencing patching.....	147

C

Cancel button	292
class	
adding	
attribute	66
file 296	
instance	73
adding from scratch	57
attribute types	
connection.....	64
expression.....	63
method.....	64
variable.....	63
connection types	64
copying.....	55
definition	18
deleting	72
editing	61
class connection	64
Class Editor	56
class information, table of properties.....	61
class instance, definition.....	18
class type	
component	63
configuration	63
none.....	63
policy	63
values	63
Client properties	
execute	65
protect	64

component	
adding to a package	296
editing	287
component class instances	193
component classes	193
examples	110
filtering	193
component data, replacing	292
compound names	
prefixes	50
using for instances	50
connection attribute type	64
connections	
drag-and-drop for component class instances ...	110
enabling	216
context menus	53, 267
Copy Package and Components dialog box	242
copying package	242
Current Feature Settings	233
customer support	4
customizing Radia	19

D

DATA1.MSI	239
Database	<i>See</i> Radia Database
Default Verify Options	124
defer notification	182
Delete button	291
Disable feature	235
distribution model	
description	17
elements	
application files	17
applications	17
deployment destination	17
deployment source	17
users	17
domain	
adding	54
definition	18
drag-and-drop connections	
avoiding errors	100

component class instances	110
drag-and-drop Notify	175

E

Edit Instance dialog box	36
Edit this Component	
description	286
File Re-Promote dialog box	290
when to use	284
editing a component	287
Editing Class dialog box	34, 61
Editing Instance dialog box	75
Editor Selector dialog box	289
buttons	291
using	291
Editors list box	289
Email notification	174
event reporting	129
keywords	130
Exist/Exist	255
express filters	
customizing	194
description	194
expression attribute type	63
Extended Verify Option	124

F

Feature Set Editor	
Add/Remove Programs	239
Current Settings	233
dialog box	226
Disable feature	235
Features menu	229
Hide feature	235
Ignore	232
Installed on First Use	230
navigating the tree	227
Not Available	232
shortcut menu	225
Features menu	229
file	
adding	296

adding a domain	54
definition	18
File Re-Promote dialog box	290
Files and Progress dialog box	157
Filter Instances dialog box	189, 193
filter specifications	
description	188
examples table	188
valid expressions	188
filtering	187
filters	
permanent	
deleting	191
description	188
setting	188
temporary	
description	191, 192
setting	191, 192

H

Hide and Disable shortcut menu	235
Hide feature	235

I

Ignore	232
IGNORE feature state	232
includes connection	64
Info/Rename button	292
inheritance	52
base instance	52
Installed on First Use	231, 233
instance	
adding	73
adding from scratch	74
compound names	50
copying	73
deleting	83
editing	75
prefixed names	50
renaming	82
instance names, prefixes	50
Instance Options tab	35

K

KB AutoImport tab	39
-------------------------	----

L

LICENSE file	44
Lock the splitter bar distance to the right side of the screen	34
logon dialog box	23

M

Manager properties, table	65
mandatory applications	126
Master Groups, creating	167
Master service	165
MD5 signature	145, 148
menu-based Notify	175
message digest algorithm	145
method attribute type	64
mouse	
table of actions	22
using	22
MSI packaged products	165
MSIEXEC	245
MSIFEATS	225, 243, 245
MST	245
multiple instances	
editing	77
Editing Instance dialog box	78
selecting	77

N

NAME attribute	18
Native Mode	245
New Application Wizard	120
variables	125
Not Available	232
notify	
by machine name	40
specify STARTDIR	40
specify UID	40
Notify	

creating an audience	176
description	174
drag-and-drop	175
file structure	186
initiating	174
menu-based selection	175
remove an application	179
retry	186
Retry Manager	187
send an e-mail	178
Status window	184
unsuccessful attempt	186
update an application	179
Notify tab	40
NOVADIGM domain	45

O

OK button	292
Optimize Service	147
configuring	149
using	147
optional applications	126
Override Value dialog box	259
owning package	193

P

package, duplicating	242
patch building	159
PATCH class	45, 145, 162
PATCH domain	45, 145, 162
patch-building restrictions	158
permanent filters	
deleting	191
description	188
setting	188
POLICY domain	45
Populate Windows Installer Methods check box ...	124
pop-up menus, how to open	267
prefixes, using in compound names	50
PRIMARY file	18
ADMIN domain	45
domains of	46

NOVADIGM domain	45
PATCH domain	45
POLICY domain	45
SOFTWARE domain	45
SYSTEM domain	45
PRIMARY file, description	45
PROFILE file, description	45
Prune Below	48

R

Radia

customizing	19
mouse actions	22
terminology	17

Radia Database

adding	
class	55
copying	55
from scratch	57
class attribute	66
domain	54
file 296	
instance	73
copying	73
from scratch	74
attribute	18
attribute value	18
changing tree view	46
class	18
class instance	18
contents of	44
deleting a class	72
deleting an instance	83
domain	18
drop down lists	69
editing	53, 267
class	61
instance	75
file	18
filtering	187
flag attribute	67
flag set attribute	68
hierarchy	24

instance	18
Manager Properties table	65
navigating through tree structure	46
navigation history	48
discarding	48
Prune Below	48
optional editing	67
PRIMARY file	18
renaming an instance	82
structure	17
SYSTEM domain	18
tree view	24
USER class	18
Version Groups	
adding a Version instance	138
assigning a Version instance	141
deleting a Version instance	144
editing	144
removing a Version assignment	144
Radia Integration Server	264, 265
Radia Inventory Manager	264
Radia MSI methods	125
Radia Registry File Editor ... <i>See</i> Registry File Editor	
Radia Service Groups	165
Master Service	165
Radia System Explorer	
capabilities	16
context menus	53, 267
control features	32
database, changing tree view	46
filtering the database	187
navigating the tree structure	46
opening	22, 200
options	32
advanced options tab	37
instance options tab	35
KB AutoImport options tab	39
notify options tab	41
pop-up menus	267
window appearance	25
re-sizing	25
radskman	165

Registry File Editor	250
accessing	250
dialog box	252
exiting the dialog box	260
navigating the tree	253
registry key, editing	256
Registry Resource instance	250
Rename Instance dialog box	82
Replace Component Data	
description	292
when to use	284
requires connection	64
resolution process	
description	200, 210
evaluating stop expressions	208
RIMSFSCAN	265
RIMWBEM	265
Run all from My Computer	230
Run all from Source	231
Run from My Computer	230
Run from Source	230

S

security dialog box	23
Service Groups, creating	165
Service Optimization	145
SHOWINCP variable	239
SOFTWARE domain	45, 47
Software Management client	264
Specify a domain name	181
stop expressions	208
SYSTEM domain	18, 45
System Explorer	<i>See</i> Radia System Explorer

T

technical support	4
temporary filters	
deleting	192
description	191, 192
setting	191, 192
terminology in Radia	17
tool tips, description	24

transform file.....	245
transient classes.....	62
tree view, manipulating.....	46

U

Unconditional/Enforced	255
Unconditional/Exist	255
USER class	45
description	18
USERID attribute	18

V

variable attribute type	63
Verify Options	124
Version Group	
adding a Version instance	138

assigning a Version instance	141
deleting a Version instance.....	144
editing	144
removing a Version assignment	144
Version Group Editor, description	134

W

WBEM namespace.....	265, 266
---------------------	----------

Z

ZOBJPTCH variable.....	145
ZRSCDATE attribute	290
ZRSCSIZE attribute	290
ZRSCTIME attribute.....	290
ZSERVICE instance	45