

# HP OpenView Administrator Workstation Using Radia

for the Windows operating system

Software Version: 4.0

---

## System Explorer Guide

Document Release Date: January 2007



## 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 1998-2007 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.

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

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

## Support

Please visit the HP OpenView support web site at:

**<http://www.hp.com/managementsoftware/support>**

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

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

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and log in. Many also require a support contract.

To find more information about access levels, go to:

**[http://www.hp.com/managementsoftware/access\\_level](http://www.hp.com/managementsoftware/access_level)**

To register for an HP Passport ID, go to:

**<http://www.managementsoftware.hp.com/passport-registration.html>**

# Preface

## About this Guide

### Who this Guide is for

The System Explorer Guide for the HP OpenView Administrator Workstation Using Radia (System Explorer Guide) is for systems administrators who want to create and maintain a distribution model for their enterprises.

### What this Guide is about

This guide introduces the HP OpenView System Explorer Using Radia—the desktop-enterprise configuration and management tool of the HP OpenView Administrator Workstation Using Radia suite—and describes the concepts of maintaining a distribution model in the Radia database, and the actions you take when administering the Radia environment.

# Revisions

This printing of the System Explorer Guide for Windows is for use with HP OpenView Administrative Workstation Using Radia version **4.0**. It contains the following changes to information and procedures for the following chapters.



Items with bullet are specific to version 4.0 or 4.0.1.

## Chapter 2: Getting Started

- 4.0.1** Page 38, Table 6: Help Menu Options: deleted the row for "Search For Help On". This option was removed from the Help menu.
- 4.0** Page 40, Setting Preferences: The Options Dialog Box. Added Notify to the list of tabs on the View Options dialog box.
- 4.0** Page 41, Table 8: Functions of the General Tab Options. Added a row for the new window resizing option, "Lock the splitter bar distance to the right side of the screen."
- 4.0** Page 50, Notify Tab: this is a 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 3: Radia Database Overview

- 4.0.1** Page 54, The Radia Database: modified the description of the PRIMARY file for the Radia Database versions 4.0 and 4.0.1:
  - Added Table 12 to summarize the PRIMARY file default domains.
  - Added default domains for CLIENT and PRDMAINT.
  - Added a note explaining the PRDMAINT domain was previously named the NOVADIGM domain, and an existing NOVADIGM domain will remain for self-maintenance of clients at versions prior to 4.0.

- Added Table 13 to describe the PRIMARY file optional domains, including the MACHINE, NOVADIGM, OS, USAGE and PATCHMGR domains.
- Page 57, added text: If you have installed HP OpenView Server Management Using Radia you will encounter the following PRIMARY file domains: APPMGMT, BASELINE, and CFM. Refer to the Radia documentation for more information on these domains and classes.

**4.0** Page 40, Setting Preferences: The Options Dialog Box. Added Notify to the list of tabs on the View Options dialog box.

## Chapter 4: Building a Database Distribution Model

**4.0.1** Page 103, Table 17: updated the list of POLICY classes in the Radia Database versions 4.0 and 4.0.1:

- Added the optional classes used with the OS Manager product: Manufacturer (MANUFACT), Machine Models (MODEL), Machine Roles (ROLE), and Machine Subnets (SUBNET).
- Added the Multicast (MULTICAST) class for use with an HP OpenView Multicast Server Using Radia.
- Added the Mobile Device Config (MBLCONFIG) class for use with wireless PDAs being managed by the HP OpenView Mobility Server Using Radia.

## Chapter 5: Additional Features of System Explorer

**4.0** Page 194, 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.

4.0 Page 148, Defining MSI Basic Behaviors on the New Application Wizard: replaced Figure 50 showing the New Application Wizard with MSI Basic Behaviors section.

## Chapter 8: Registry File Editor

Page 271, Managing Feature Sets in Native Mode: corrected text to read:  
"The MSI native behavior will be used for update and repair."

## Chapter 9: Registry File Editor

- 4.0 Page 280, Table 30: Registry File Editor Context Menu Options: added rows defining Delete Always, Delete Never and Delete (Default), which apply when you perform a Radreg32 Delete operation from the Registry File Editor.

## Chapter 10: Support for the Inventory Manager

- 4.0.1** Page 290, About the Inventory Manager: revised topic to discuss how the Messaging Server and Radia Reporting Server enhance the collection and reporting of Inventory data in a Radia environment.
- 4.0** Page 292, About the Reporting Server: new topic summarizes the dynamic and intuitive reporting capabilities of the web-based HP-OpenView Reporting Server Using Radia.
- 4.0** Page 294, CIM Schema and Inventory Collection: new topic explains how Radia audit collection uses the industry-standard CIM schema.

## Appendix C: Modifying Component Instances (Non-Production Only)

- 4.0** Page 325, Editing a Component: Added a caution. The component editor is not aware of a component's target operating system. After an edit session, you may need to restore the original values of attributes such as ZRSCMMEM (PDS Member Name), ZRSCCRC (Resource CRC), ZRSCPADM (PDS AdminID), and ZPUBVER if the change is unintentional.



## 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.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 Might also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE Might also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

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

**Table P.3 ~ Terminology\***

Term	Might also be called
Application	software, service
Client	Application Manager and/or Software Manager
Computer	workstation, server

**Table P.3 ~ Terminology\***

Term	Might also be called
NOVADIGM domain	PRDMAINT domain  Note: The PRDMAINT domain handles the maintenance functions for Client version 4.0 or above. The NOVADIGM domain handles the maintenance functions for Client versions prior to 4.0.  Therefore, if you have Clients in your database that are 2.x or 3.x levels, you will see the NOVADIGM domain in the database as well as the PRDMAINT domain.
Configuration Server	Manager, Active Component Server



# Contents

- Preface ..... 5
  - About this Guide .....5
    - Who this Guide is for .....5
    - What this Guide is about .....5
  - Revisions .....6
    - Chapter 2: Getting Started .....6
    - Chapter 3: Radia Database Overview .....6
    - Chapter 4: Building a Database Distribution Model .....7
    - Chapter 5: Additional Features of System Explorer .....7
    - Chapter 9: Registry File Editor .....8
    - Chapter 10: Support for the Inventory Manager .....8
    - Appendix C: Modifying Component Instances (Non-Production Only) .....8
      - Editorial Improvements .....9
  - Conventions .....10
  
- 1 Introduction ..... 21
  - About the System Explorer for the HP OpenView Administrator Workstation Using Radia .....22
  - The Distribution Model .....23
  - HP OpenView Using Radia Terminology .....24
  - Handling Your Unique Requirements .....25
  - Summary .....27

<b>2 Getting Started .....</b>	<b>29</b>
Using a Mouse to Manipulate Elements in the Window .....	30
Opening the System Explorer .....	31
The System Explorer Window .....	33
Basic Windows Operations .....	33
Customizing the System Explorer Window .....	34
System Explorer Window Menus .....	34
The System Explorer Toolbar .....	39
Setting Preferences: The Options Dialog Box .....	40
General Tab .....	41
Instance Options Tab .....	44
Advanced Tab .....	47
KB AutoImport Tab .....	48
Notify Tab .....	50
Summary .....	52
<b>3 Radia Database Overview .....</b>	<b>53</b>
The Radia Database .....	54
Navigating the Radia Database .....	59
Navigation History .....	61
Compound Names .....	63
Inheritance in the Radia Database .....	65
System Explorer Shortcut Menus .....	66
Working with Domains .....	66
Adding a Domain .....	66
Working with Classes .....	67
Adding a Class .....	67
Copying a Class .....	67
Create a New Class .....	70
Editing a Class .....	72

Client Properties.....	78
Manager Properties.....	78
Adding an Attribute to a Class.....	80
Optional Attribute Editing Controls.....	82
Deleting a Class.....	86
Working with Instances.....	88
Adding an Instance.....	88
Copying an Instance.....	88
Adding an Instance.....	89
Editing an Instance.....	90
Editing Multiple Instances.....	92
Renaming an Instance.....	97
Deleting an Instance.....	98
Summary.....	100
<b>4 Building a Radia Database Distribution Model.....</b>	<b>101</b>
Building a Distribution Model.....	102
The Policy Domain.....	102
Software Domain.....	107
Inter-Package Relationships.....	110
Determining Policies.....	112
Connection Types.....	115
Manual Connections.....	115
Drag-and-Drop Connections.....	116
Show Connections.....	119
Drag-and-Drop Component Class Instances.....	126
Connecting Services to Policies.....	128
A Word about the Policy Server.....	132
Summary.....	133
<b>5 Additional Features of the System Explorer.....</b>	<b>135</b>
Features of the System Explorer.....	136

Running the New Application Wizard .....	136
Defining MSI Basic Behaviors on the New Application Wizard.....	147
The Version Group Editor.....	149
Creating a Version Group Instance .....	149
Version Group Assignments .....	156
Service Optimization .....	160
Using the Optimize Service Option .....	163
Radia Service Groups .....	178
Component Enhancements.....	179
Prerequisites .....	179
Creating a Master Service for a Service Group .....	180
Radia Notify .....	187
Notify Dialog Boxes.....	189
Send an E-mail Using Notify .....	192
Update an Application Using Notify .....	193
Remove an Application Using Notify .....	193
Set Notification Details .....	194
To defer notification until a specific date and time.....	196
Notify File Structure.....	199
Retrying a Notify Operation.....	201
Drag-and-Drop Notify for Wake-On-LAN Clients .....	201
Filtering—Viewing Portions of the Radia Database .....	202
Permanent Filters .....	203
Temporary Filters .....	206
Viewing the Base or Null Instance .....	207
Appending Filter Results.....	207
Filtering Component Class Instances.....	208
Express Filters .....	210
Customizing Express Filters.....	210
Building State Files with the System Explorer.....	214
State File Creation Process .....	216
Summary.....	221



<b>6</b>	<b>Understanding the Resolution Process .....</b>	<b>223</b>
	The Resolution Process.....	224
	Heaps and Objects.....	224
	Client Explorer .....	224
	Resolution Process—Where Does It Begin? .....	226
	Resolution Example.....	229
	Resolution Process and Symbolic Substitution .....	230
	Summary.....	234
<b>7</b>	<b>Maintaining Connection Rules in the Radia Database .....</b>	<b>235</b>
	Introduction .....	236
	Connection Rules Overview.....	236
	Enabling Connections in New Classes .....	239
	Summary.....	246
<b>8</b>	<b>Feature Set Editor .....</b>	<b>247</b>
	Feature Set Editor Overview .....	248
	Accessing the Feature Set Editor .....	249
	About the Radia MSI Feature Set Editor .....	249
	Tree View.....	250
	Features Menu.....	252
	Current Feature Settings.....	257
	Hide and Disable Features .....	259
	Feature Set Editor Buttons .....	262
	Package Information.....	262
	Maximum Size .....	262
	Total Size.....	263
	Selected Size .....	263
	Windows Add/Remove Programs Applet.....	263

Copying a Windows Installer Enabled Package .....	268
Managing Feature Sets in Native Mode .....	271
Summary .....	273
<b>9 Registry File Editor .....</b>	<b>275</b>
The Registry File Editor.....	276
Accessing the Registry File Editor .....	276
Expanding and Collapsing the Tree.....	279
Set Properties Context Menu.....	280
Editing a Registry Key .....	282
Closing the Registry File Editor .....	285
Summary.....	287
<b>10 Support for the Inventory Manager.....</b>	<b>289</b>
About the Inventory Manager .....	290
Overview of Auditing.....	291
About the Reporting Server .....	292
Radia and WbEM.....	293
CIM Schema and Inventory Collection .....	294
Summary.....	295
<b>A Using System Explorer Help .....</b>	<b>297</b>
Getting Help.....	297
Traditional Microsoft Windows-style Help.....	297
Context-Sensitive Help .....	297

B System Explorer Shortcut Menus .....	305
C Modifying Component Instances (Non-Production Only) .....	321
When to Use these Commands .....	322
Accessing the Add, Edit, and Replace Component Commands.....	323
Editing a Component .....	324
Using the Editor Selector Dialog Box.....	329
Replacing Component Data .....	330
Adding a Component.....	334
Summary .....	338
Index .....	339



---

# 1 Introduction

At the end of this chapter, you will:

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

# About the System Explorer for the HP OpenView Administrator Workstation Using Radia

The System Explorer for the HP OpenView Administrator Workstation Using Radia (System Explorer) is an interactive graphical tool for manipulating and inspecting the content of the Radia database.

By working in the System Explorer, you can:

- Define users (that is, 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 that are being managed.  
(Example: a user's user ID)
- **Applications**  
The software that is being managed.  
(Example: Microsoft Office 2000)
- **Application Files**  
The components that make up the application.  
(Example: Microsoft Office 2000 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)
- **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 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.

# HP OpenView Using 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**. Refer to the HP OpenView using Radia publication library for additional information.

**Table 1: HP OpenView Using Radia Terminology**

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>Database Reference Manual for HP OpenView Using 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	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



Term	Short Description
	traditional flat file. The attributes of a class instance object contain data describing one specific entity of that class or instance.
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 Attribute Value	An attribute is a data element (also called a field or property) 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.
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 <b>John Doe</b> , and the USERID attribute contains the value <b>JDOE</b> .

## Handling Your Unique Requirements

Radia ships the Radia database with a pre-configured 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 the Radia-supplied components, and integrating your own in-house developed or third-party-supplied executables.

In particular, this document describes the System Explorer using the Radia database as it is configured when shipped from HP. If your organization customizes Radia, the System Explorer can be used as described here, but the

sample screens shown in this document might 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 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, relationships, and intended configurations of the computers in your enterprise being managed by Radia.
- Although Radia ships with a pre-configured database, the framework is highly customizable.



---

## 2 Getting Started

At the end of this chapter, you will:

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

# 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 HP OpenView System Explorer Using Radia (System Explorer). With its familiar graphical user interface, the System Explorer supports standard mouse actions activated by clicking or double-clicking the left mouse button. 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: Summary of Mouse Actions**

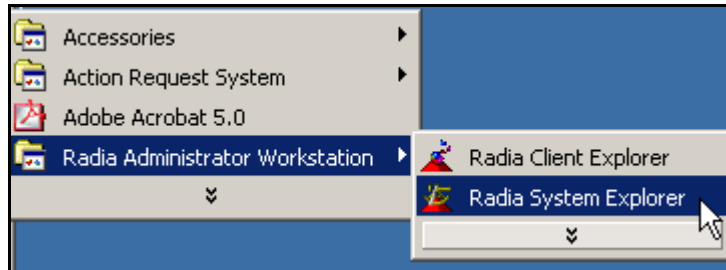
<b>Mouse Action</b>	<b>Result</b>
Click the left mouse button. (We refer to this as a left-click, or simply a click.)	Highlight an icon, select a drop-down menu command, or press a button.
Double-click the item using the left mouse button. (We refer to this as choosing an item.)	Open the next level window 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 System Explorer to configure your enterprise desktop environment, you will use many familiar elements: windows, drop-down menus, shortcut menus, menu commands, dialog boxes, and so forth.

# Opening the System Explorer

To open the System Explorer

- 1 Click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**.

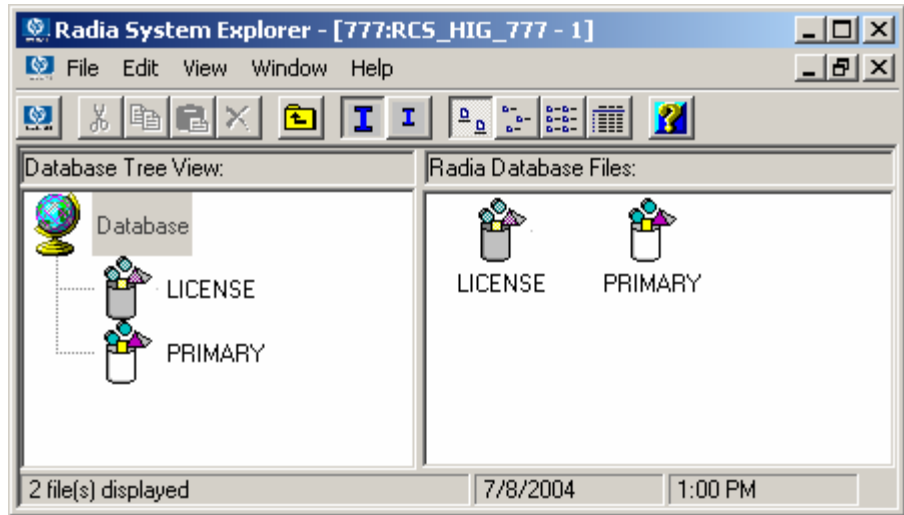


A logon dialog box opens.



- ▶ The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might have been changed during installation.  
You can 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 initial Radia System Explorer window opens.



The System Explorer has a familiar appearance, using a multiple document interface (i.e., you can open multiple System Explorer windows simultaneously). Beneath the title bar, there is a drop-down menu, a tool bar, and a workspace containing System Explorer windows. Each 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 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 the Radia Database* on page 59. With the Database icon selected in the tree view, the list view of the System Explorer window displays a list of the files comprising the Radia database.

The 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

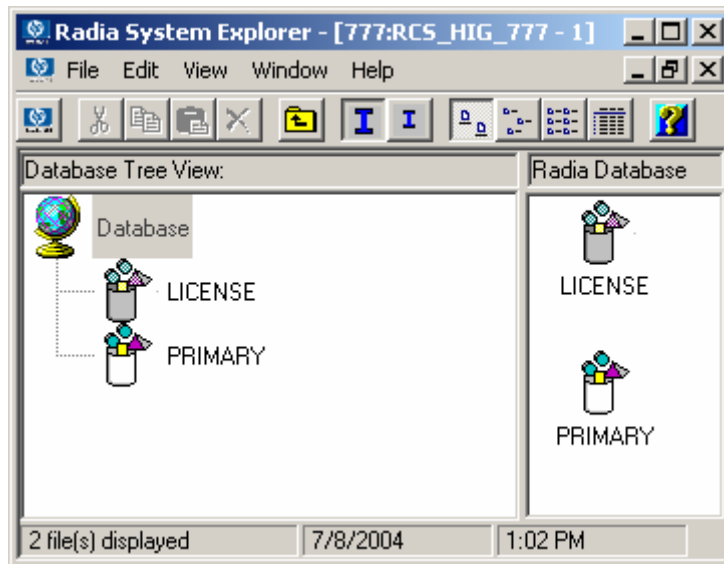


spontaneously when the mouse pointer is moved over such an item in the System Explorer window.

## The System Explorer Window

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

### Basic Windows Operations



**Figure 1: 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 System Explorer window by clicking and dragging the frame between them.

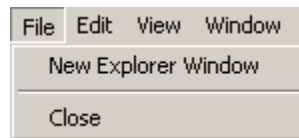
## Customizing the System Explorer Window

You can manipulate and customize the way the 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 System Explorer Toolbar](#) on page 39 for additional information about these functions.

### System Explorer Window Menus

- **The File Menu**

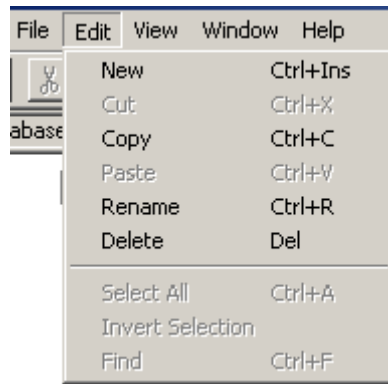
Use the options in this window to open additional System Explorer windows or to close the current window. Multiple windows can be open at the same time. You can drag items between multiple windows.



**Figure 2: 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 3: The Edit menu.**

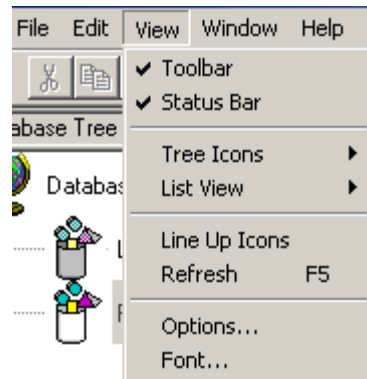
**Table 3: Edit Menu Options**

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

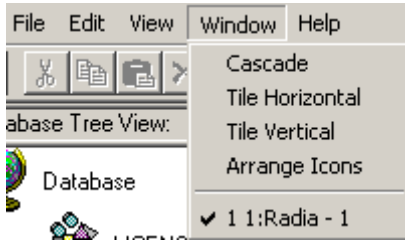


**Figure 4: The View menu.**

**Table 4: View Menu Options**

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.
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 System Explorer window, lining up all of the elements for easy reading.
Options	Opens the Radia System Explorer Options dialog box. See Setting Preferences: The Options Dialog Box on page 40 for more information.
Font	Opens a dialog box that allows you to change the font used to display text in the System Explorer windows.

- **The Window Menu**  
Use the options in this window when you have more than one System Explorer window open.

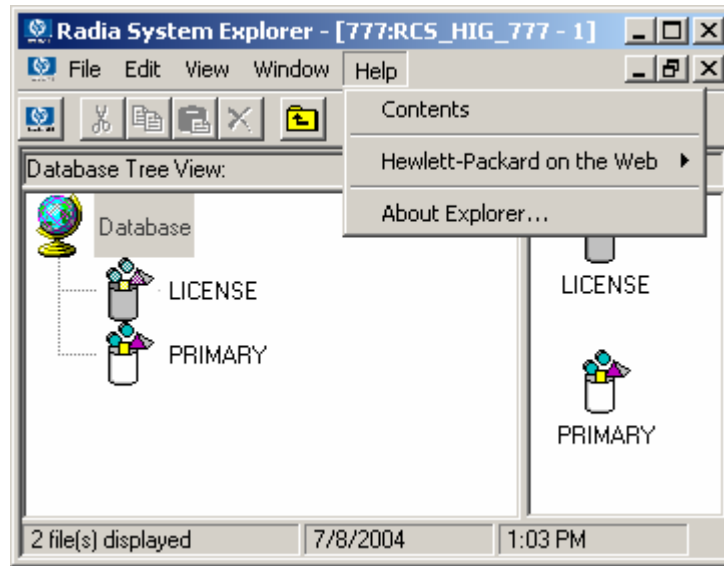


**Figure 5: The Window menu**

**Table 5: Window Menu Options**

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

- **The Help Menu**  
Get details about the System Explorer or link to the HP OpenView web site.



**Figure 6: The Help menu.**

**Table 6: Help Menu Options**

Option	Description
Contents	Opens a traditional Windows-style help interface. See Using System Explorer Help on page 297 starting on page 297.
Hewlett-Packard on the Web	Opens a sub-menu with links to the Hewlett-Packard web site pages for Technical Support and Home Page. Selecting one of these links opens your default Web browser and connects to the selected page over the Internet.
About Explorer	Displays information about the current System Explorer—version number, Configuration Server (Manager) Information, and so forth. Click <b>System Info</b> to display a standard Microsoft System Information window. This application analyzes your machine and shows the results in an easy to navigate format.

## The System Explorer Toolbar













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




**Figure 7: The System Explorer toolbar.**

The following table describes the functions of the toolbar buttons.

**Table 7: System Explorer Toolbar Button Functions**

Button	Function
	Open a new 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 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 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 System Explorer window at one time.
	Display the list view vertically. Adds a column for TYPE (such as Class and Instance). Maximizes the number of items and provides detailed description of

Button	Function
	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 System Explorer provides an Options dialog box to allow you to control some of the features of the System Explorer.

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

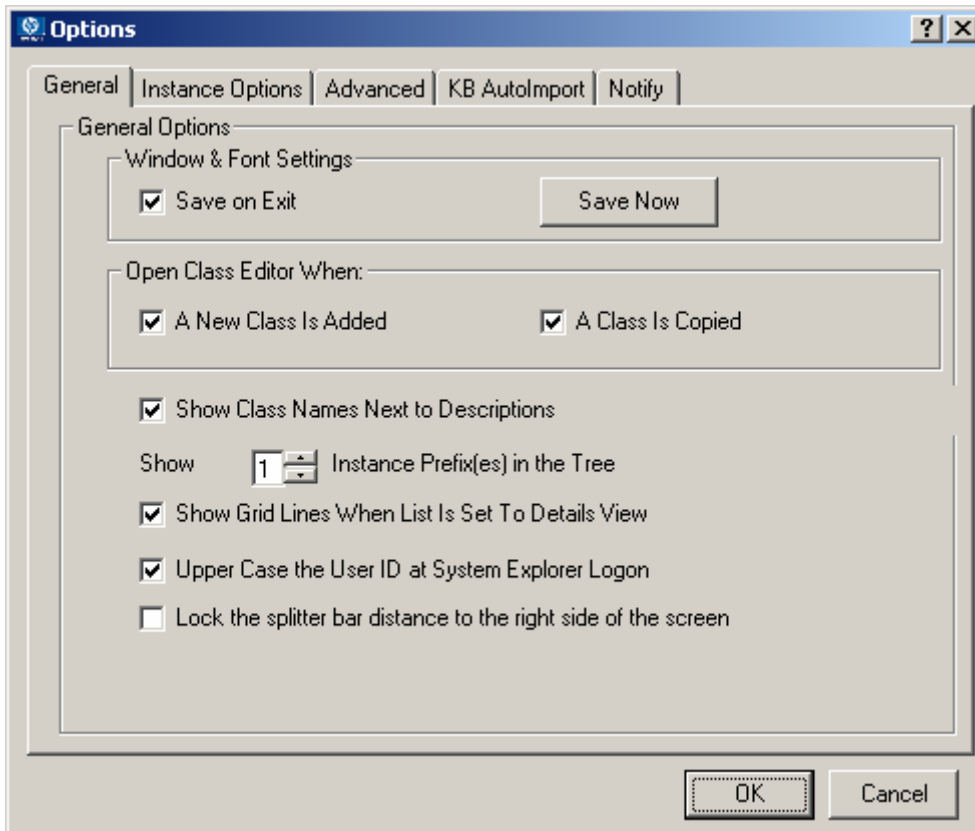
There are five tabs on the Options dialog box:

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

- Click a tab to access its configuration options.



## General Tab



**Figure 8: The General tab of the System Explorer Options dialog box.**

Table 8 describes the functions of the options on the **General** tab.

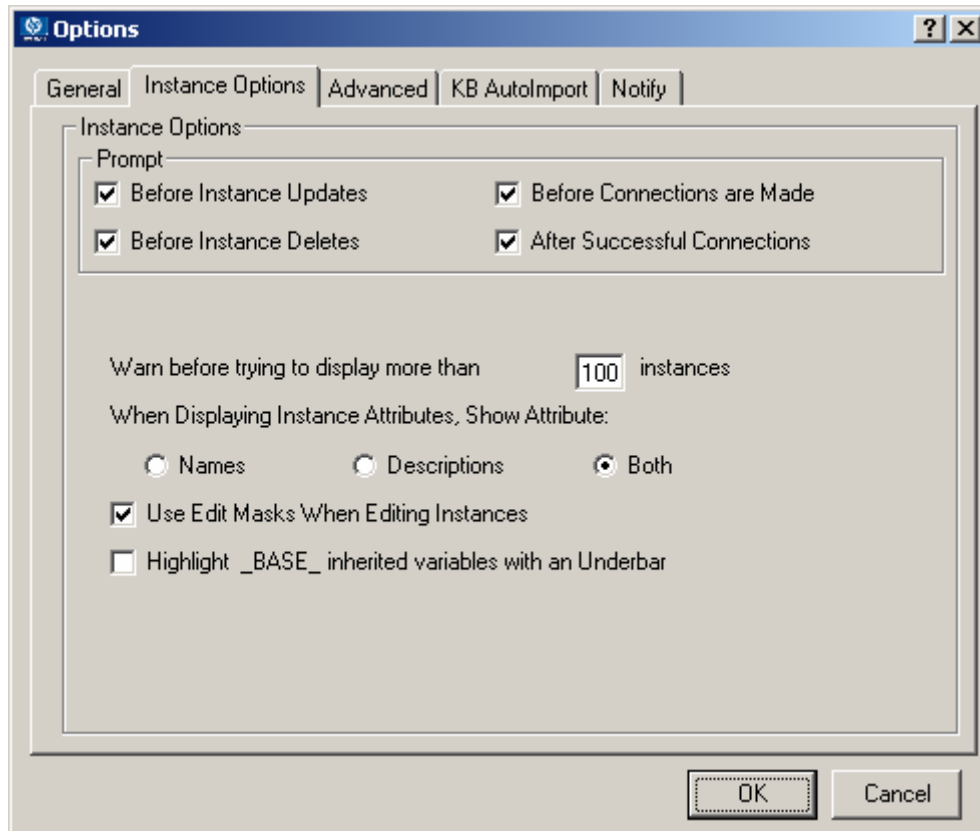
**Table 8: Functions of the General Tab Options**

Option	Function and Use
Window & Font Settings	Select <b>Save on Exit</b> to have the System Explorer track certain window settings and the font to use for displaying text when you exit from the System Explorer session.  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 System

Option	Function and Use
	<p>Explorer window, and the relative width of each pane. Window settings are saved independently for each level of the Radia database.</p> <p>Click <b>Save Now</b> to have System Explorer save the current window settings and font in use.</p> <p>You can change the font System Explorer uses to display text by choosing a font from the dialog box that opens when you click <b>View</b> → <b>Fonts</b> from the menus.</p>
Open Class Editor When	<p>Select the <b>A New Class is Added</b> check box to have System Explorer open the Editing Class dialog box when you add a class.</p> <p>Select the <b>A Class is Copied</b> check box to have System Explorer open the Editing Class dialog box when you copy a class.</p> <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 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 System Explorer window. The Details View in the list view of the System Explorer window always shows the internal name for classes in the Type column.</p>
Show <i>n</i> Instance Prefix(es) in the Tree	<p>This option can be set to 0, 1, 2, or 3 to control how many levels of instance name prefixes (delimited by the underscore character) will be compressible and expandable in the tree view. This is described in Navigating the Radia Database on page 59.</p>
Show Grid Lines When List Is Set To Details View	<p>Select this check box to have 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 System Explorer Logon	<p>Select this check box to automatically convert the user ID into all uppercase.</p>

<b>Option</b>	<b>Function and Use</b>
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 System Explorer window during resizing. This is useful if you are focusing on information on the right side of the 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

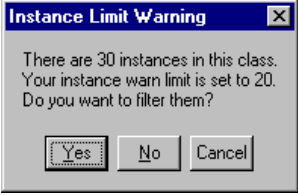



**Figure 9: The Instance Options tab of the System Explorer Options dialog box.**

Table 9 describes the functions of the options on the **Instance Options** tab.

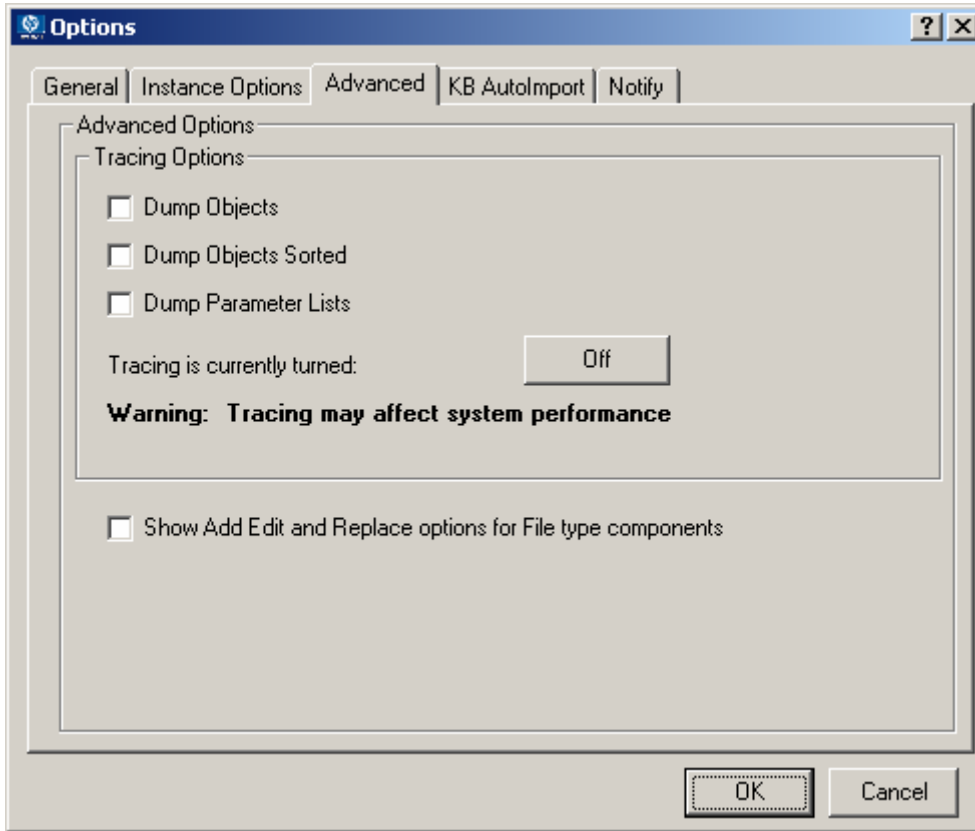
**Table 9: Functions of the Instance Options Tab Options**

Option	Function and Use
Prompt	These four check boxes control whether 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.

Option	Function and Use
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>  <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 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"> <li>• Click <b>Yes</b> to open the Filter Class dialog box, and enter an appropriate filter.</li> <li>• Click <b>No</b> to display all of the instances.</li> <li>• Click <b>Cancel</b> to close the warning without displaying any instances.</li> </ul>
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 System Explorer window, when you open an instance by double-clicking its name in the tree view.</p>

<b>Option</b>	<b>Function and Use</b>
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 82.</p>
Highlight <code>_BASE_</code> 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 10: The Advanced tab of the System Explorer Options dialog box.**

Table 10 describes the functions of the options on the **Advanced** tab.

**Table 10: Functions of the Advanced Tab Options**

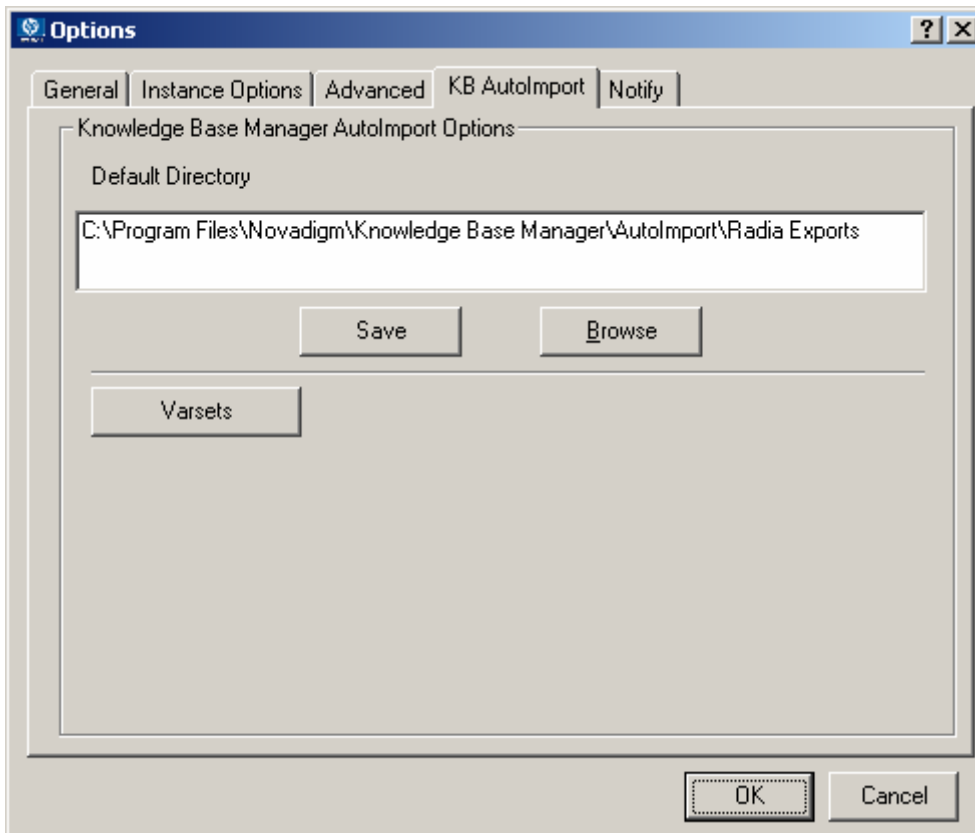
Option	Function and Use
Tracing Options	<p>These settings are only used for troubleshooting the System Explorer.</p> <p>▶ Do not change these settings unless instructed to do so by HP Technical Support.</p>

Option	Function and Use
Show Add, Edit and Replace options for File type components	<p data-bbox="524 216 1398 275">WARNING: This setting must be used in non-production environments only.</p> <p data-bbox="524 296 1443 357">This setting determines whether to display the following commands on the shortcut menus for file-type components:</p> <ul data-bbox="565 378 915 506" style="list-style-type: none"> <li data-bbox="565 378 857 407">• Edit this Component</li> <li data-bbox="565 428 915 457">• Replace Component Data</li> <li data-bbox="565 478 824 508">• Add a Component</li> </ul> <p data-bbox="524 527 1281 556">For details, see Appendix C, Modifying Component Instances.</p>

## KB AutoImport Tab

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





**Figure 11: The KB AutoImport tab of the System Explorer Options dialog box.**

### To specify the KB AutoImport options

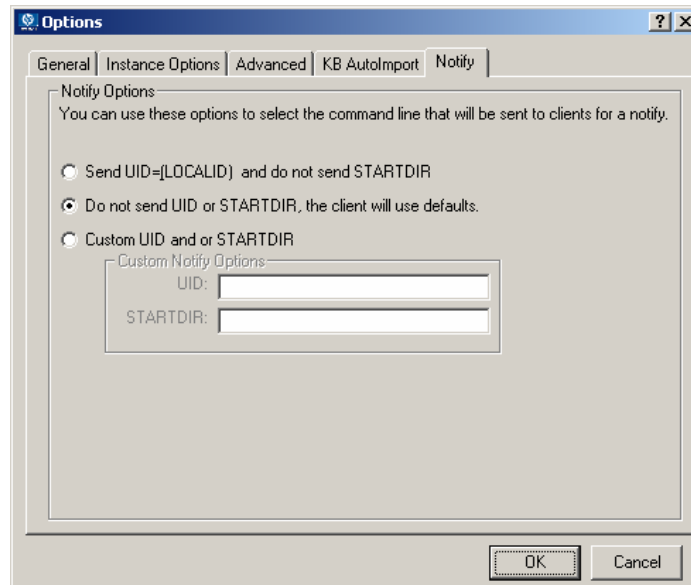
- 1 Specify or browse to the default directory to be used as an export directory when you build **state files** for import into the Knowledge Base Manager database. The default location is the `\Radia Exports` directory that was installed by the Knowledge Base Manager. This directory includes the required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

See Building State Files with the System Explorer on page 214 for details on building state files from application services.

## Notify Tab

Use the **Notify** tab to specify the `radskman.exe` command line values for UID and STARTDIR 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 Notify Manager and displayed on the Notifications Details dialog box as shown in Figure 54 on page 194. For more information, see the topics Radia Notify on page 187, Notify Dialog Boxes on page 189, and Set Notification Details on page 194.



**Figure 12: The Notify tab of the System Explorer Options dialog box.**

Table 11 describes the functions of the options on the **Notify** tab.

**Table 11: Functions of the Notify Tab Options**

<b>Option</b>	<b>Function and Use</b>
Send UID=[LOCALID] and do not send STARTDIR	<p>Creates a Notify command with UID=[LOCALID]. This command notifies clients by the name of the user currently logged on.</p> <p>To notify clients by the name of the machine, use the <b>Custom UID and or STARTDIR</b> option.</p>
Do not send UID or STARDIR, the client will use defaults.	<p>Creates a Notify command without any values for UID or STARTDIR; clients use the default values for UID and STARTDIR.</p> <p>To specify non-default values, use the <b>Custom UID and or STARTDIR</b> option.</p>
Custom UID and or STARTDIR	<p>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 machine name, or to specify the appropriate starting directory.</p> <ul style="list-style-type: none"><li>• Accept the <b>Custom UID</b> 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.</li><li>• Accept the <b>Custom STARTDIR</b> value of SYSTEM to notify and install an application in the machine context. You can also enter another STARTDIR value.</li></ul>

## Summary

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

---

## 3 Radia Database Overview

At the end of this chapter, you will:

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

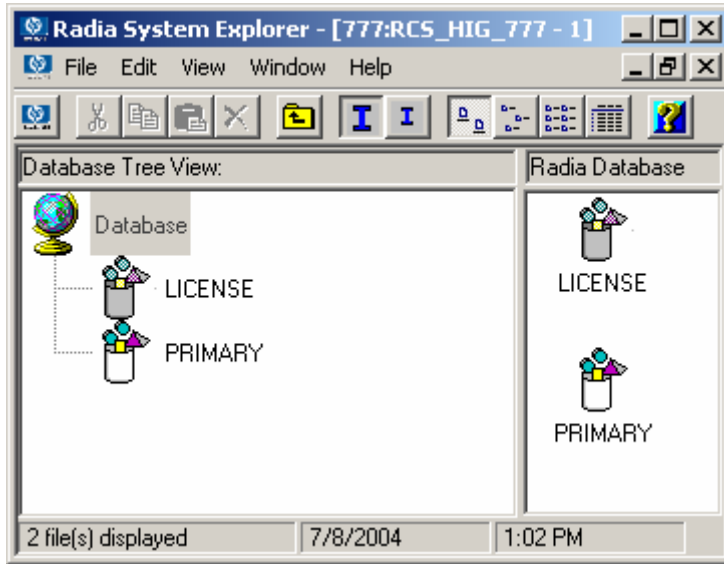
## The Radia Database

The Radia database, located on the HP OpenView Configuration Server Using 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 System Explorer to view and manipulate the Radia database.

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



**Figure 13: The Radia database—LICENSE and PRIMARY files.**

- The LICENSE file is read-only and used for 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 eight default domains, as discussed in Table 12 below.

For more information on the PRIMARY file domains and their classes and attributes, please refer to the *Database Reference Manual for the HP OpenView Configuration Server Using Radia (Database Reference Manual)*.

**Table 12: PRIMARY File Default Domains**

Domain	Description
ADMIN	Use the ADMIN domain to limit administrative access to the Radia database, and specify rules for connecting classes.

<b>Domain</b>	<b>Description</b>
AUDIT	Use the AUDIT domain to configure tasks that will inventory client computers' assets. Refer to the <i>Installation and Configuration Guide for the HP OpenView Inventory Manager Guide (Inventory Manager Guide)</i> for additional information.
CLIENT	Use the CLIENT domain to configure Client Operations Profiles, including setting access points for application data. Refer to the <i>Installation and Configuration Guide for the HP OpenView Application Manager Using Radia (Application Manager Guide)</i> and the <i>Installation and Configuration Guide for the HP OpenView Software Manager Using Radia (Software Manager Guide)</i> for additional information.
PATCH	Use the PATCH domain to store the list of available patches within a PATCH class. See Service Optimization on page 160 for additional information about this domain.
POLICY	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. Refer to the Implementing Entitlement Policy chapter in the <i>Application Manager Guide</i> or the <i>Software Manager Guide</i> for more information
PRDMAINT	Use the PRDMAINT domain to store packages for self-maintenance of Radia Clients at versions 4.0 and above. Refer to the <i>Software Manager Guide</i> for more information on self-maintenance.  <b>Note:</b> Prior to Radia Database version 4.0, the PRDMAINT domain was named the NOVADIGM domain. Use an existing NOVADIGM domain to store packages for self-maintenance of Radia Clients at versions prior to 4.0.
SOFTWARE	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.
SYSTEM	Use the SYSTEM domain to store administrative and process control definitions.

Table 13 on page 57 describes the optional domains of the PRIMARY file. These optional domains can be selected for activation during the Configuration Server installation, or may exist from a prior version of the Radia Database.



In addition to the optional domains, if you have installed HP OpenView Server Management Using Radia you will encounter the following PRIMARY file domains:

- APPMGMT
- BASELINE
- CFM

Refer to the Radia product documentation for more information about these domains and their classes and attributes.

**Table 13: PRIMARY File Optional Domains**

Domain	Description
MACHINE	Use the MACHINE domain to define templates for the machines being managed by the HP OpenView Operating System Manager Using Radia (OS Manager) product. Refer to the <i>Installation and Configuration Guide for the OS Manager (OS Manager Guide)</i> for more information.
NOVADIGM	Use an existing NOVADIGM domain to store packages for self-maintenance of Radia Clients at versions prior to 4.0. <b>Note:</b> This domain was renamed the PRDMAINT domain as of version 4.0.
PATCHMGR	Use the PATCHMGR domain with the HP OpenView Patch Manager Using Radia (Patch Manager) product. Refer to the <i>Installation and Configuration Guide for the Patch Manager</i> for more information.
OS	Use the OS domain to define the operating system services to be deployed to target machines with the OS Manager product. Refer to the OS Manager Guide for more information.
USAGE	Use the USAGE domain with the HP OpenView Usage Manager Using Radia (Usage Manager) product. See the <i>Installation and Configuration Guide</i> for the Usage Manager for more information.

- 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 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 request 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 187.

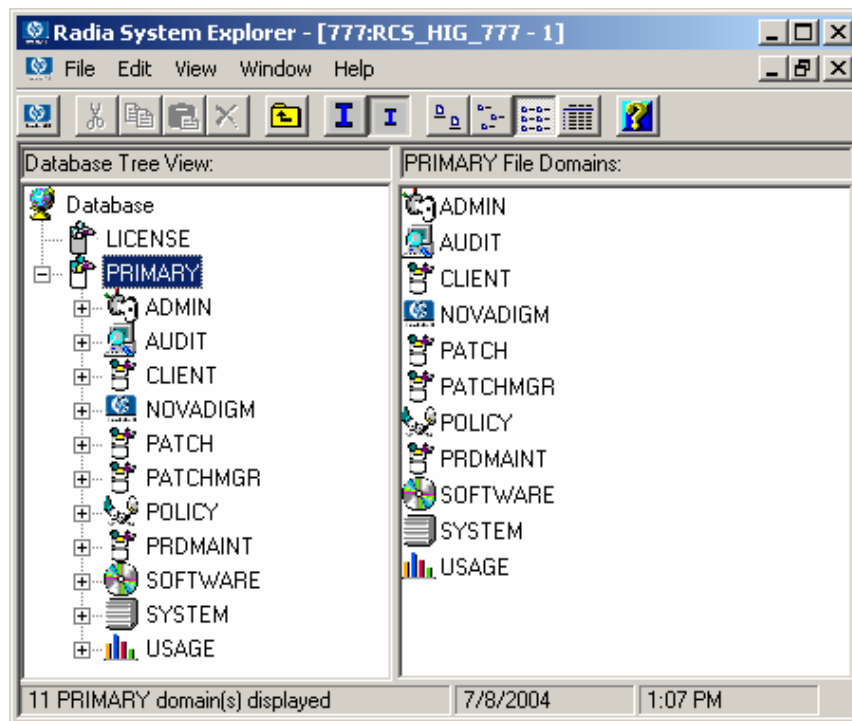
# Navigating the Radia Database

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 the tree structure

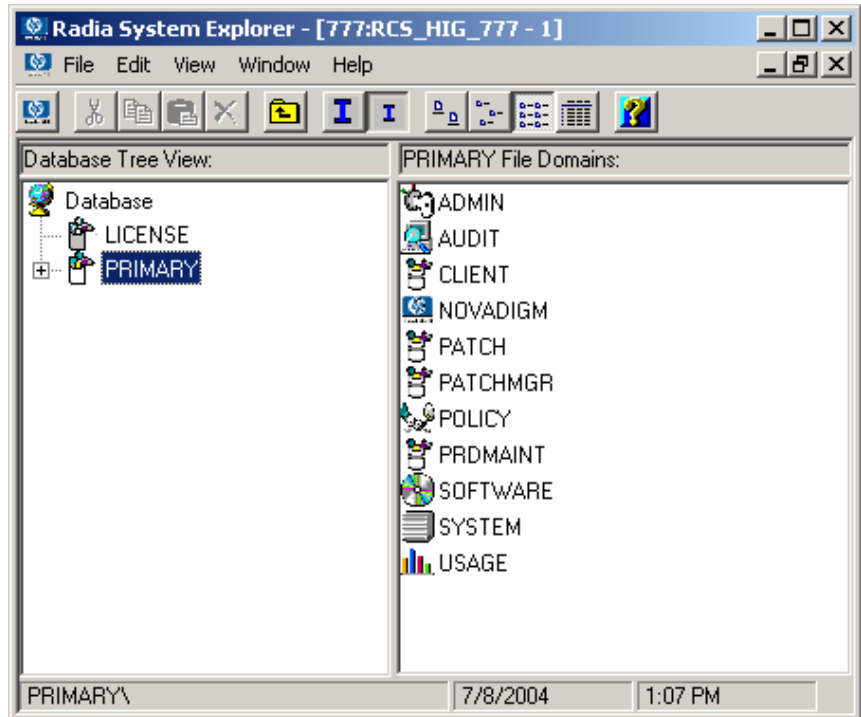
- 1 Open the System Explorer, and double-click the **PRIMARY** file in the tree view.

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



The 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 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.



- 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 System Explorer window. The classes of the SOFTWARE domain will be displayed in the list view of the 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 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.

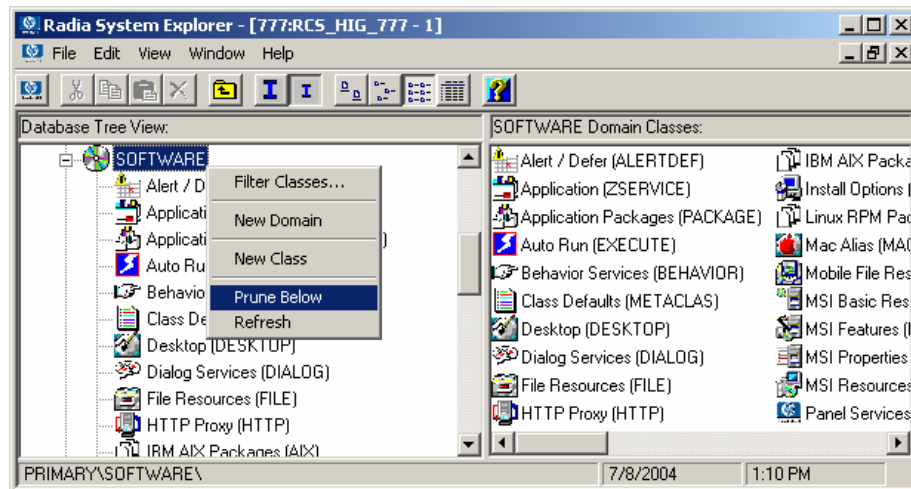
## Navigation History

As you open and close components in the 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 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.



- 2 Click **Prune Below**.

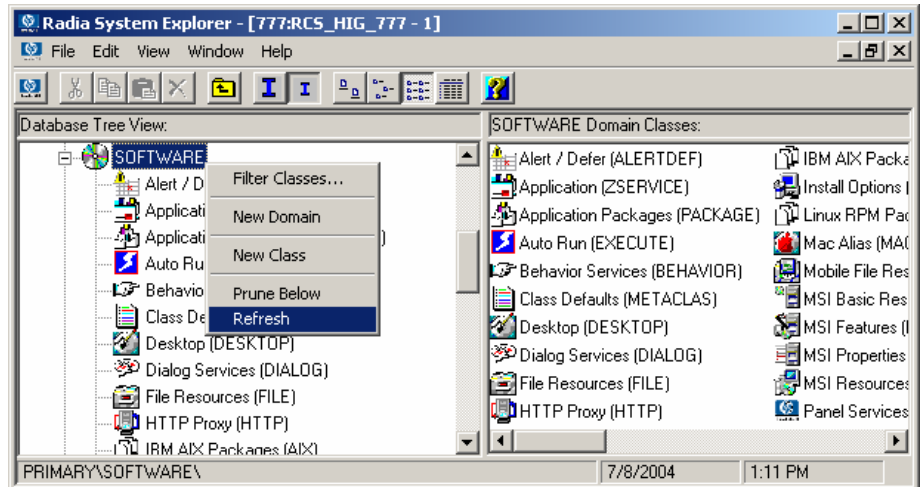
The tree view collapses all open nodes below the SOFTWARE domain, and discards the history of their having been open.

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.



- 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.

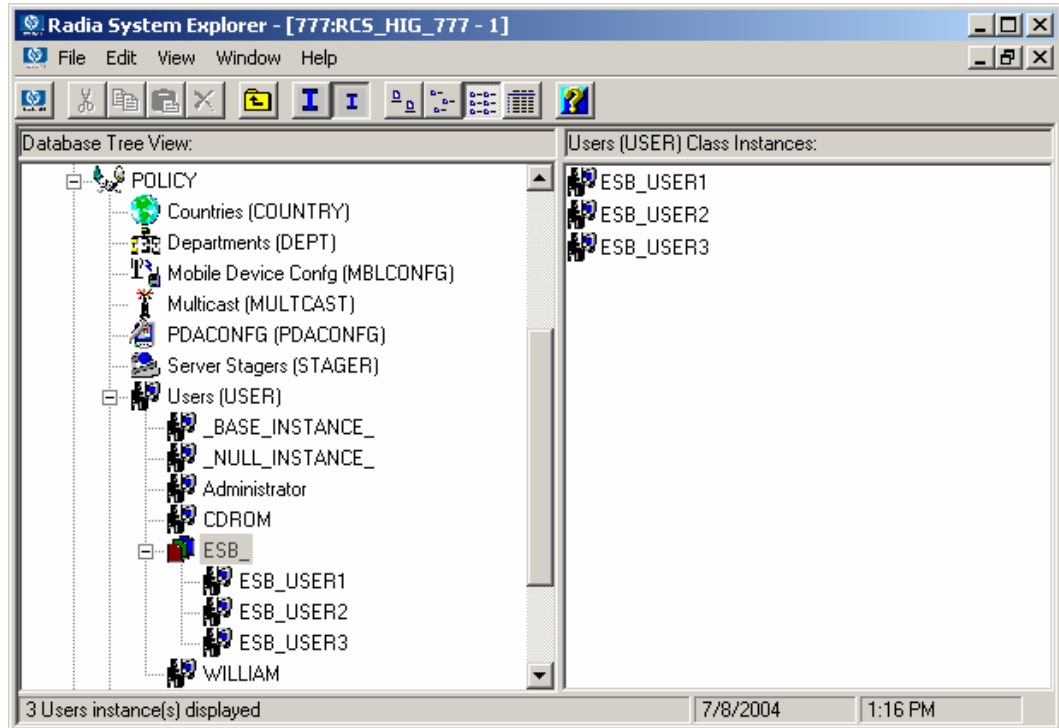
## Compound Names

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 202). 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 14: Compound names being used to manage large numbers of instances.**

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

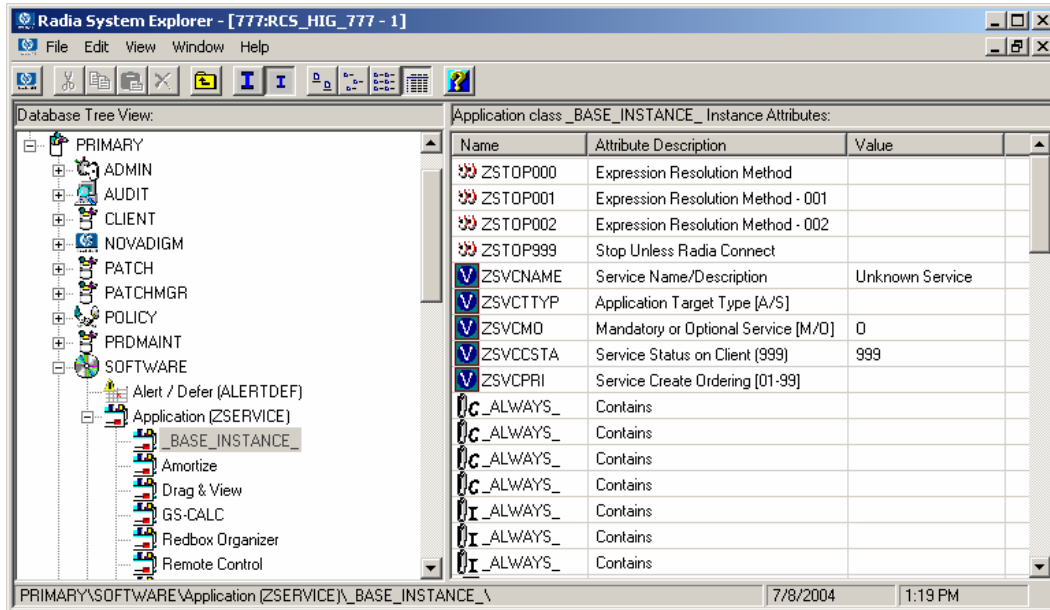
The number of prefix levels in an instance name (0, 1, 2, 3) that are displayed in the tree view is controlled by the **Show n Instance Prefixes in the Tree** option setting. This is found on the **General** tab of the Options dialog box and can be accessed from the **View** menu. The Options dialog box is described in Setting Preferences: The Options Dialog Box starting on page 40.

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



# Inheritance in the Radia Database

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



**Figure 15: The `_BASE_INSTANCE_` of the Application (ZSERVICE) class.**

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 67 to learn how), and the value re-appears in the instance automatically, the value is being inherited from the `_BASE_INSTANCE_`.

## System Explorer Shortcut Menus

In the System Explorer you can add, copy, edit, rename, and delete Radia database components (domains, classes, and instances). It also provides extensive support for connecting instances.

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

For a complete list of the shortcut menus, see Appendix B, System Explorer Shortcut Menus.

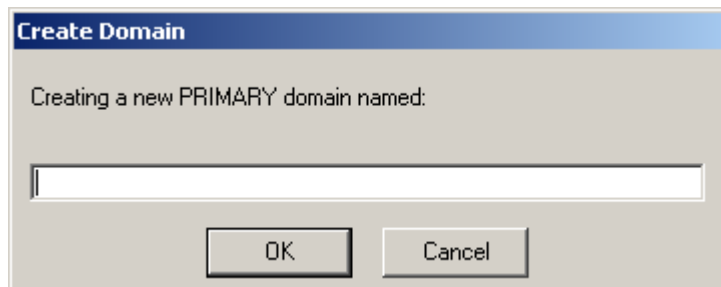
## Working with Domains

### 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.



- 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
- creating a new class

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, create a new class.

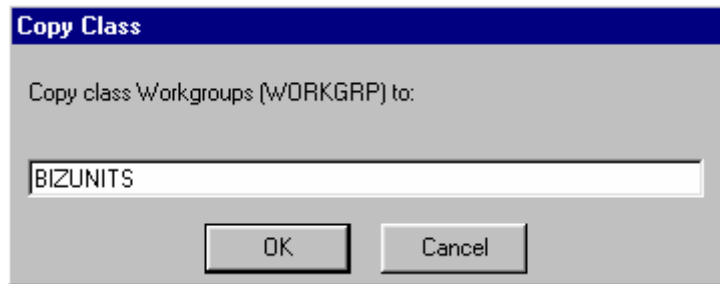
Copying a class creates a new class with the same attributes as the copied class, and a `_BASE_INSTANCE_` for the new class with its attribute values set as in the `_BASE_INSTANCE_` of the copied class.

### Copying a Class

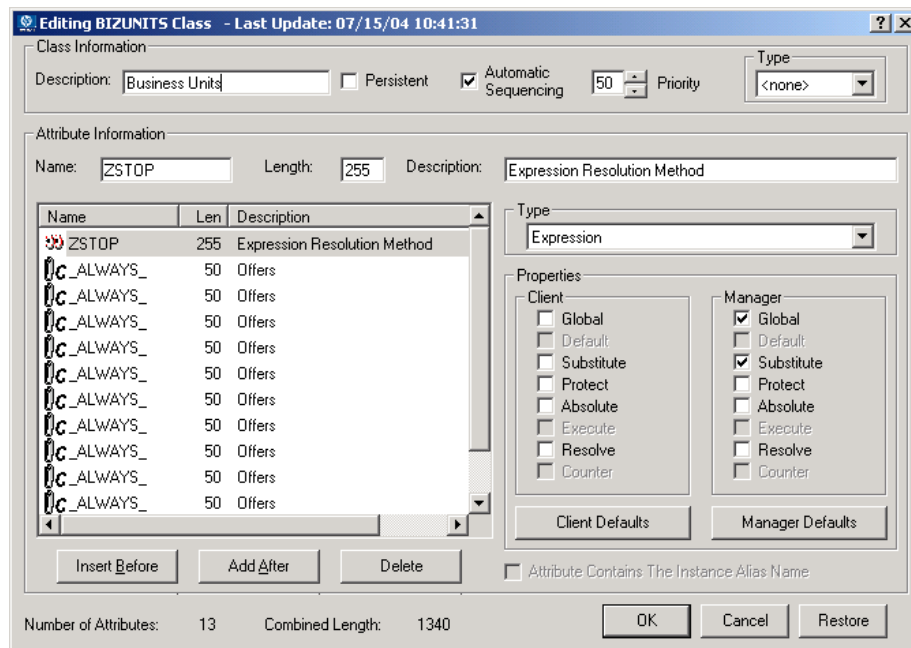
#### To copy an existing class

- 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.

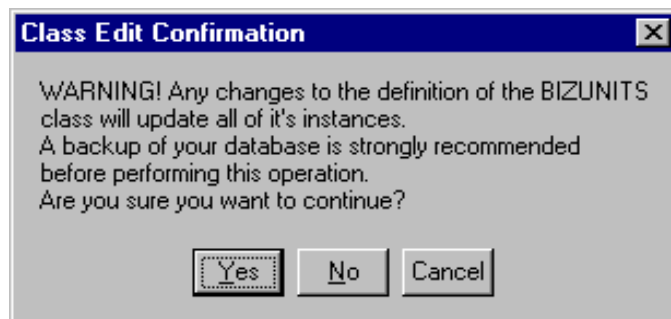


- 4 Enter a name for the new class (BIZUNITS in this example), and click **OK**.
  - If the option to open the Class Editor when a class is copied is not in effect (see Setting Preferences: The Options Dialog Box on page 40 for details), the original class is copied with the new name, the Class Editor is not opened, and no warning (see the Class Edit Confirmation warning message on page 69) is issued. You will have to manually open the Class Editor to edit the new class.
  - If the option to open the Class Editor is in effect, the Editing Class dialog box opens. See Editing a Class on page 72.



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

The Class Edit Confirmation warning message opens.



- Click **Yes**.

The new class addition is complete.

- ▶ Clicking **No** or **Cancel** to the **Class Edit Confirmation** warning message does not terminate the copying of the class—it closes the Class Editing dialog box.

If you have created a class and want to delete it, see *Deleting a Class* on page 86.

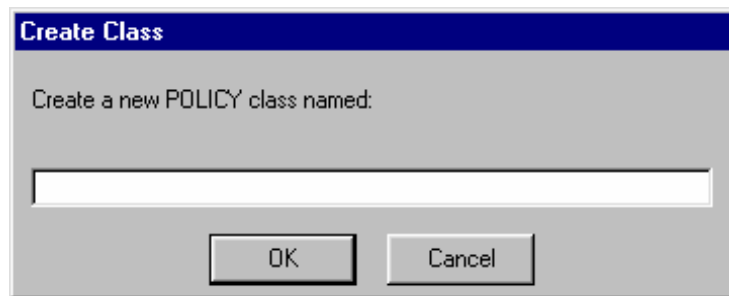
Once you have created a class, check the `MASTER_CONNECTION_RULES` variable to determine if you need to establish a drag-and-drop rule for the new class. For additional information regarding connection rules and drag-and-drop connections, see *Maintaining Connection Rules in the Radia Database* on page 235.

## Create a New Class

### To create a new class

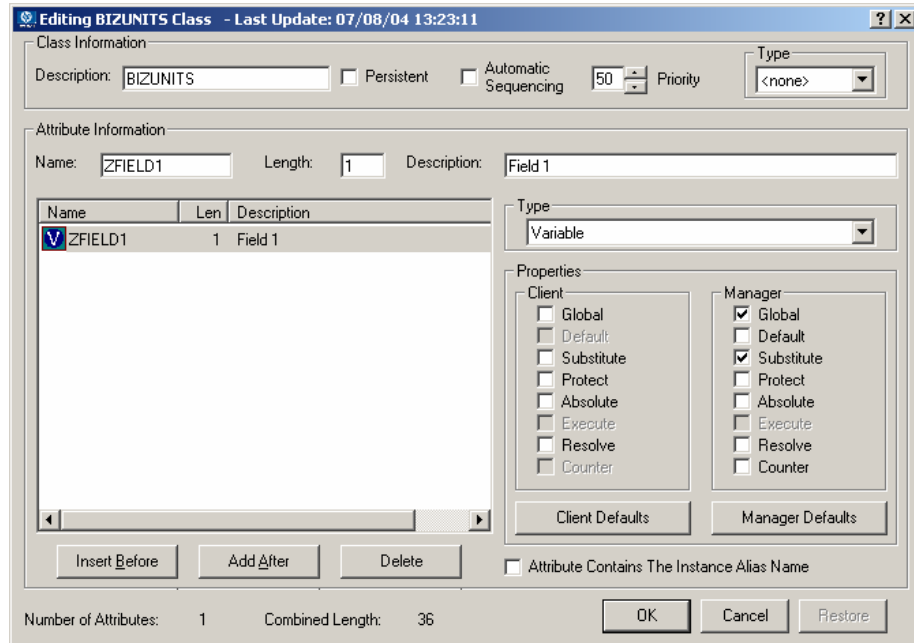
- 1 Highlight the domain where you wish to insert the new class.
- 2 Right-click the domain name.
- 3 Select **New Class** from the shortcut menu.

The Create Class dialog box opens.



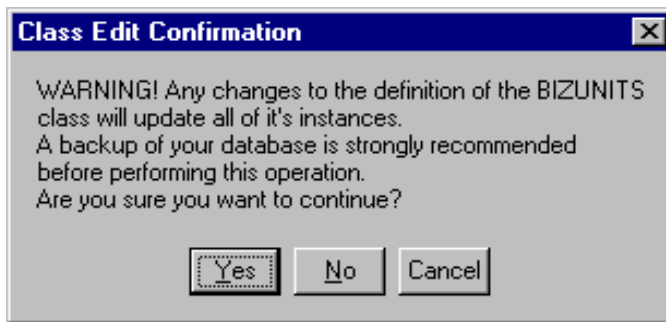
- 4 Enter a name for the new class (BIZUNITS in this example), and click **OK**.

- If the option to open the Class Editor when a class is copied is not in effect(see Setting Preferences: The Options Dialog Box on page 40 for details), the original class is copied with the new name, the Class Editor is not opened, and no warning (see the Class Edit Confirmation warning message on page 69) is issued. You will have to manually open the Class Editor to edit the new class.
- If the option to open the Class Editor is in effect, the Editing Class dialog box opens. See Editing a Class on page 72.



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

The Class Edit Confirmation warning message opens.



- 6 Click **Yes**.

The new class has been added.

▶ Clicking **No** or **Cancel** to the Class Edit Confirmation warning message does not terminate the copying of the class—it closes the Class Editing dialog box.

If you have created a class and want to delete it, see *Deleting a Class* on page 86.

Once you have created a class, check the `MASTER_CONNECTION_RULES` variable to determine if you need to establish a drag-and-drop rule for the new class. For additional information regarding connection rules and drag-and-drop connections, see *Maintaining Connection Rules in the Radia Database* on page 235.

## Editing a Class

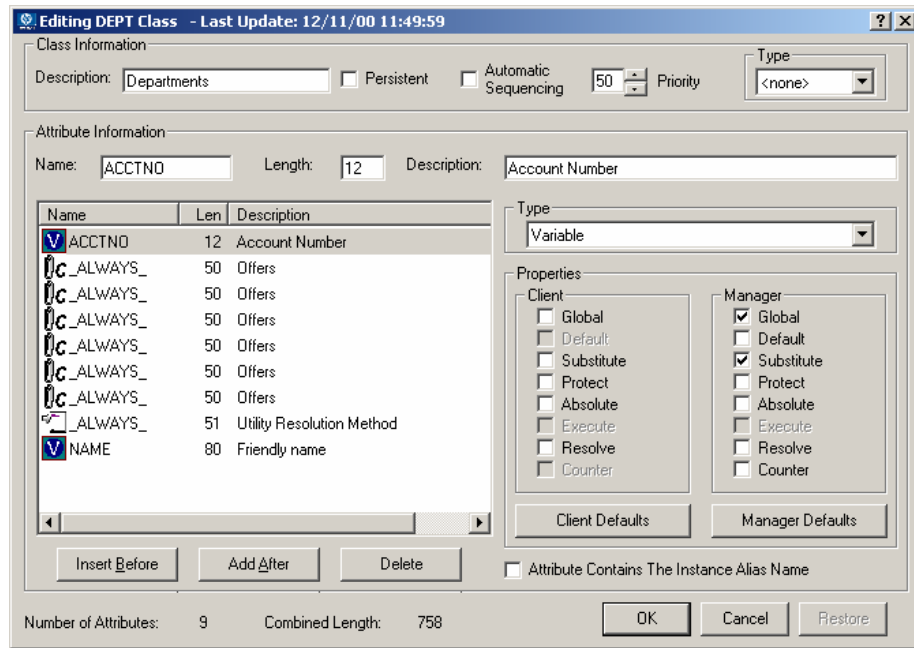
When you choose **Edit Class** from a shortcut menu, or during the addition of a new class as described in *Adding a Class* on page 67, the Editing Class dialog box opens.

### 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.





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

**Table 14: Class Information Properties**



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.




Property	Significance and Setting
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 whose 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, see Chapter 6, Understanding the Resolution Process on page 223.</p>
Automatic sequencing	<p>Normally, resolution proceeds from attribute to attribute in the order the attributes appear in the Attribute List. See Chapter 6, Understanding the Resolution Process on page 223.</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"> <li>• Expressions</li> <li>• Variables</li> <li>• Classes (connections)</li> <li>• Methods</li> </ul> <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>

Property	Significance and Setting
Priority	<p>This field is two characters long and has a value between 00 and 99. During object resolution, the Configuration Server puts this value in the ZOPJPRI variable. If there is a blank or zero value in this field, the Configuration Server supplies the default value specified for that particular class. The default values for the classes are:</p> <ul style="list-style-type: none"> <li>• PATH = 5</li> <li>• REGISTRY = 60</li> <li>• DESKTOP = 70</li> <li>• FILE = 50</li> <li>• METACLAS = 10</li> <li>• ZSERVICE = 50</li> <li>• MACALIAS = 70</li> <li>• All others = 50</li> </ul>
Type	<p>Classifies classes according to their purpose and use. 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> <p>&lt;none&gt; – This class is not classified.</p> <p><b>Configuration</b> – 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.</p> <p><b>Component</b> – This class identifies the items (files, registry entries, links, icons, etc.) that comprise the content identified by a Configuration class instance. Typically, instances of this class have data that can be distributed. Examples: FILE, REGISTRY, and DESKTOP.</p> <p><b>Policy</b> – 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.</p>

- 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 15.

**Table 15: 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.

Attribute Type	Usage
<p>Connection (available)</p>  <p>Connection (set)</p>  <p>Note: The letter in the icon denotes the connection type:</p> <ul style="list-style-type: none"> <li>• A – Always</li> <li>• C – Class</li> <li>• I – Includes</li> <li>• R – Requires</li> </ul>	<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 can be set. Set connections are attributes containing a specification for a connection to another class.</p> <p>A class connection is, in effect, a branch in the resolution process. See Chapter 6, Understanding the Resolution Process.</p> <ul style="list-style-type: none"> <li>• An Always connection is always resolved, no matter what the name of its attribute.</li> <li>• A Class connection is resolved if its attribute's name is <code>_ALWAYS_</code>, or if the name of its attribute matches the current value of the system message.</li> <li>• An Includes connection is always resolved, no matter what the name of its attribute.</li> <li>• A Requires connection blocks resolution of instances in the required instance and its children, whose instance names are identical to instance names in the requiring instance and its children.</li> </ul> <p>Requires and Includes connection types are useful, among other uses, for expressing inter-package relationships, connecting instances of the PACKAGE class.</p> <p>When one package Requires another, components of the required package are superseded by the same component in the requiring package.</p> <p>When one package Includes another, both are deployed as a single entity.</p>
<p>Method</p> 	<p>Methods are programs executed as part of the resolution process. The method attribute identifies the program to be executed.</p>

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

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

## Client Properties



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 16: 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.

Property	Significance and Setting
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 <b>Substitute</b> property determines whether symbolic substitution occurs when a variable is created as a result of a Configuration Server GET operation, such as occurs when a REXX program calls the EDMGET function. If a variable in the object retrieved by EDMGET contains a symbol, and the <b>Substitute</b> 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 expression &amp;(ZMASTER.ZUSERID) will evaluate to WILLIAM after symbolic substitution, if the value of the ZUSERID attribute of the ZMASTER object is WILLIAM at the time symbolic substitution occurs.</p>
Protect	<p>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.</p>
Absolute	<p>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.</p>
Execute	<p>When selected, indicates that the value of the attribute identifies a method to be executed by the Configuration Server.</p>
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 <b>Resolve</b> 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>

Property	Significance and Setting
Counter	<p>When selected, this check box indicates that the variable contains an integer that the Configuration Server will accumulate. The accumulated total will be stored in a variable with the same name in all parent persistent objects.</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 Configuration Server when the <b>Counter</b> 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 <b>Counter</b> 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. See Chapter 6, Understanding the Resolution Process.

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 is 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 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.

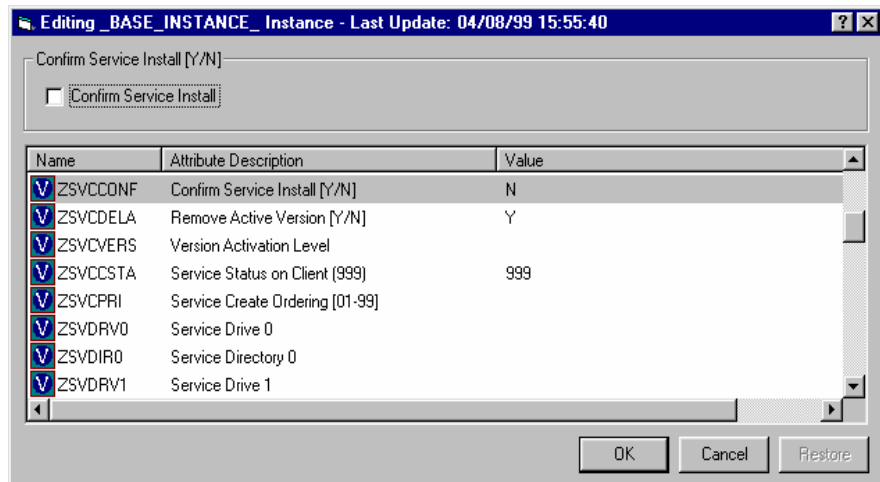
## Optional Attribute Editing Controls

The Attribute Description for variable attributes can be used to determine the control that the 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 9 on page 44 for details.

### Flag Attribute

A flag attribute is a one-character field whose value can be Y or N. To direct the 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:



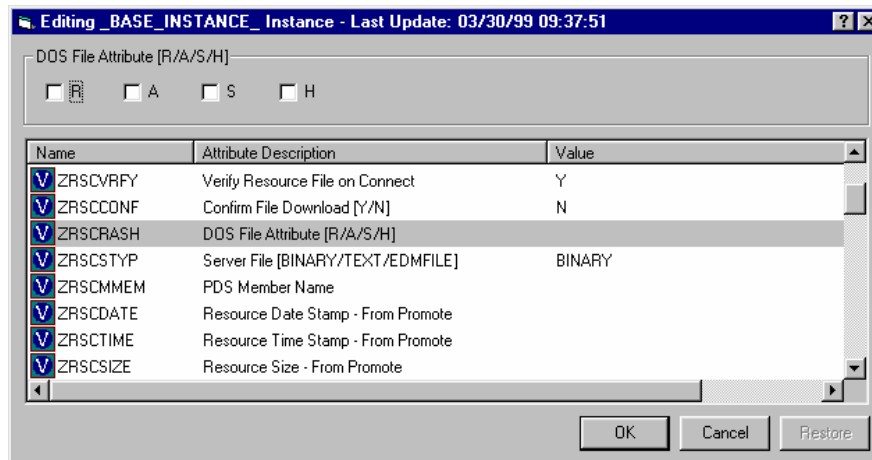
**Figure 16: The Attribute Description for the ZSVCCONF variable terminates with [Y/N], indicating a flag attribute.**

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 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, the character value that is associated with the flag appears in the flag set string. If a flag is off, the character value that is associated with the flag does not appear in the flag set string. An empty flag set string indicates that no flags are on.

For example, the DOS File Attribute flag set for a particular file is contained in the ZRSCRASH attribute of the FILE class. 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 17: An attribute terminating with one-character choices separated by slashes and enclosed in brackets is treated as a flag set.**

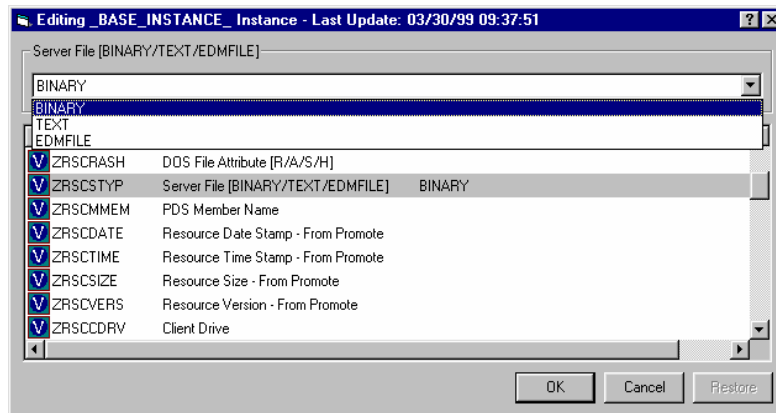
If the Attribute Description for a multi-character attribute terminates with one-character choices separated by slashes and enclosed within square brackets, and the number of choices equals the length of the attribute (as in

the ZRSCRASH attribute, above), the 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 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.

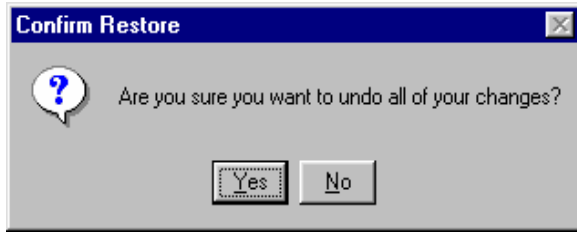
For example, the ZRSCSTYP attribute of the FILE class offers three choices in the Attribute Description, as shown here, after clicking on the drop-down list control:



**Figure 18: An attribute description that terminates with a series of choices separated by slashes and enclosed in brackets is offered a drop-down list for editing attributes.**

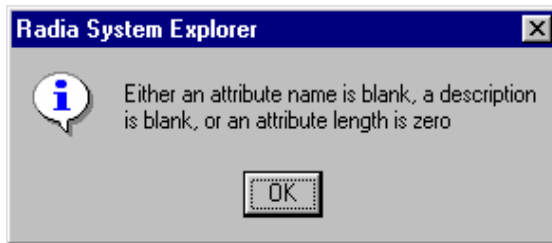
- 1 As you make changes to a class, you might change your mind and wish to start over. Once a change has been made during class editing, the **Restore** button will be enabled. Click this button to discard changes you've made to the class since the Editing Class dialog box opened.

The Confirm Restore message appears.



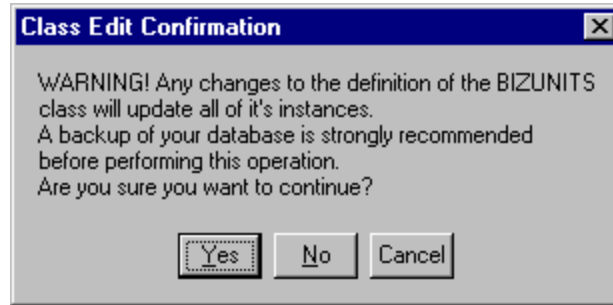
**Figure 19: Confirm Restore message.**

- Click **Yes** to discard changes and restart editing the class.
  - Click **No** to dismiss the confirmation without discarding your changes.
- 2 When you've completed editing the class, click **OK** to save your changes and exit the Editing Class dialog box.
  - 3 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 20: Warning dialog box for blank attribute fields.**

- 4 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 21: Class Edit Confirmation dialog box.**

- 5 Click **Yes** to update all existing instances of the class.

Click **No** to discard your changes to the class, and avoid updating the instances.

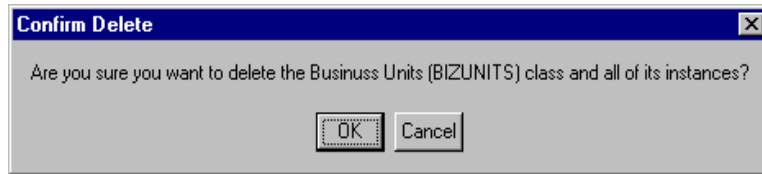
Click **Cancel** to return to the Editing Class dialog box.

## Deleting a Class

To delete a class and all of its instances

- 1 Right-click on the class in the System Explorer tree view.
- 2 Select **Delete Class** from the shortcut menu that opens.

A confirmation dialog box opens.



**Figure 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 existing instance; otherwise, add a new instance from scratch.

## Copying an Instance

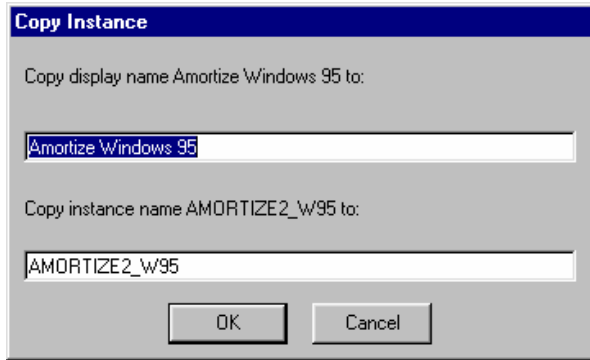
Copying an instance creates a new instance with the same attribute values as the copied instance.

### To copy an existing instance

- 1 Choose **Copy Instance** from a shortcut menu.

A **Copy Instance** dialog box opens.





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

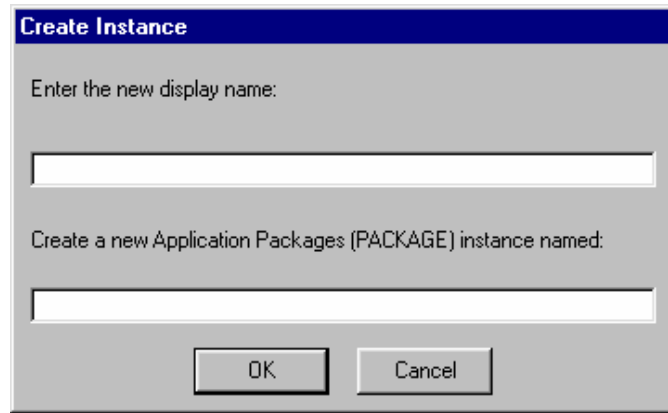
## Adding an Instance

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

### To add a new instance

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

A **Create Instance** dialog box opens.



**Figure 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 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 System Explorer window.

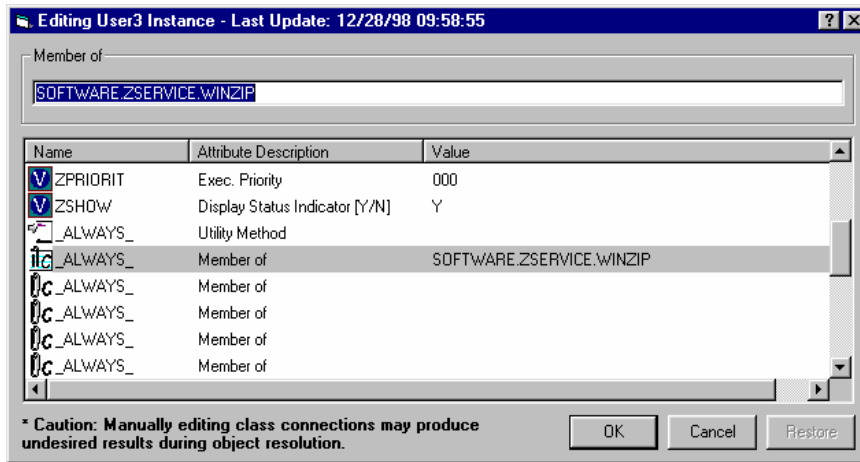
You can now edit the new instance to contain attribute values intended for the new instance.

## 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 System Explorer window.

The Editing Instance dialog box opens.



**Figure 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.

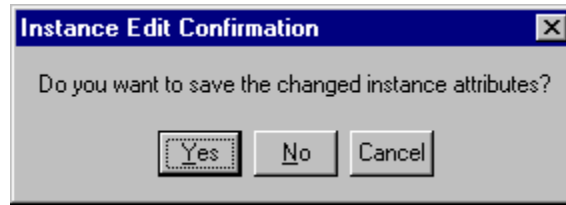
The dialog box lists the attributes comprising the instance, and their current values. The highlighted attribute is open for editing in the data entry field at the top of the dialog box.

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

See *Optional Attribute Editing Controls* on page 82 for information about how the 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 26: Instance Edit Confirmation dialog box.**

► This confirmation message opens if you have selected the **Prompt: Before Instances Updated** check box on the **Instance Options** tab of the Options dialog box (described in Setting Preferences: The Options Dialog Box starting on page 40).

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

Click **No** to discard your changes to the instance, and return to the 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 116.

## Editing Multiple Instances

The 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 System Explorer window.

Click on an instance to select it.

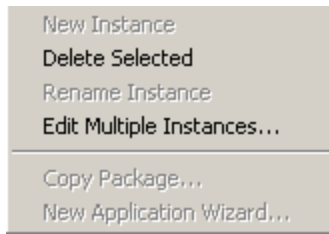
- **Ctrl**-click to add an instance to the selected set; or
- **Shift**-click and/or **Ctrl-Shift**-click to add a range of instances to the selected set.

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
Adapatibility 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

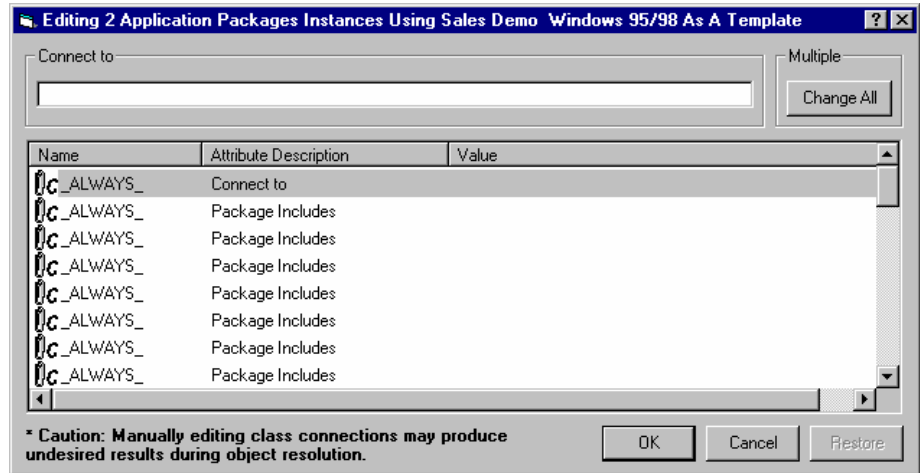
- 2 Right-click on the instance that you want to use as the template for the multiple instance-edit operation.

A shortcut menu appears:



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

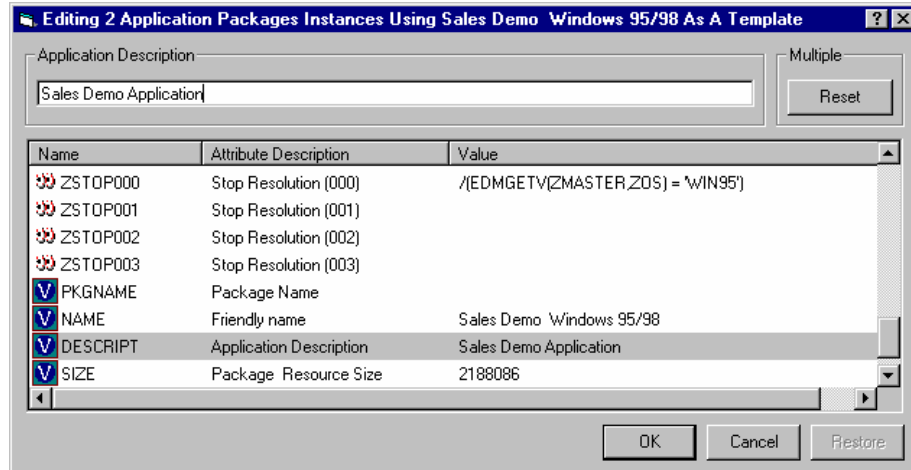
The **Editing Instance** dialog box appears:



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

- 4 Scroll to and select a variable whose value you want to copy to all instances in the selected set.
  - If the selected variable in the template instance already has the desired value, click **Change All** to copy that value to the other instances in the selected set.
  - If the selected variable does not have the desired value, edit the value.

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



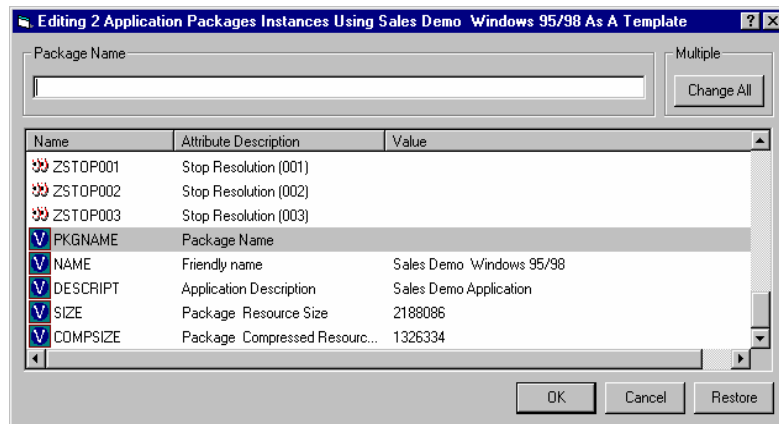
To restore the value of a currently selected variable to its original contents:

— Click **Reset**.

As you scroll through the instance'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.

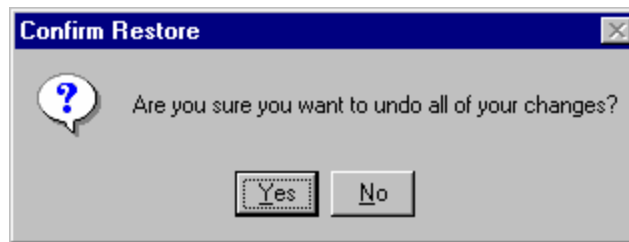
After making a change to at least one variable, and selecting a different variable, the **Restore** button becomes enabled:



To undo all changes made in the current Editing Instance dialog-box session

- 1 Click **Restore**.

You are prompted to confirm the operation:

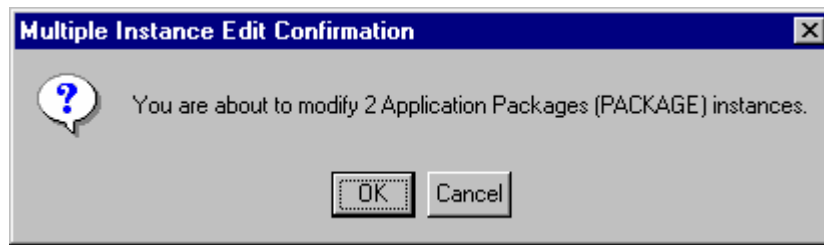


- 2 Click **Yes** to discard all changes made in the current Editing Instance dialog box session.

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:





- 3 Click **OK** to save the changes 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.

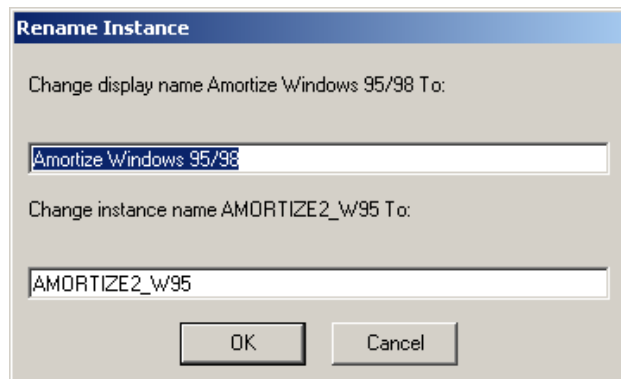
You can click **Cancel** in the Editing Instance dialog box to exit the dialog box without saving any changes to the Radia database.

## 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.



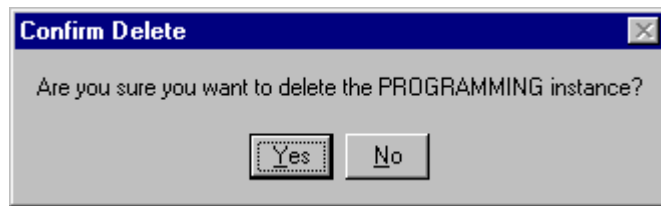
- 3 Enter a new display name (that is, a friendly name) and/or a new instance name for the instance, and click **OK**.

## Deleting an Instance

Be careful when deleting instances. Instances are connected via the connection attributes they contain. When you delete an instance, any other instance connected to the deleted instance will have a broken connection. This can produce unexpected and undesirable 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.



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 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 System Explorer shortcut menus to help you add, edit, copy, rename, and delete Radia database components.

---

## 4 Building a Radia Database Distribution Model

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 HP OpenView Policy Manager Using Radia (Policy Manager) and its uses.

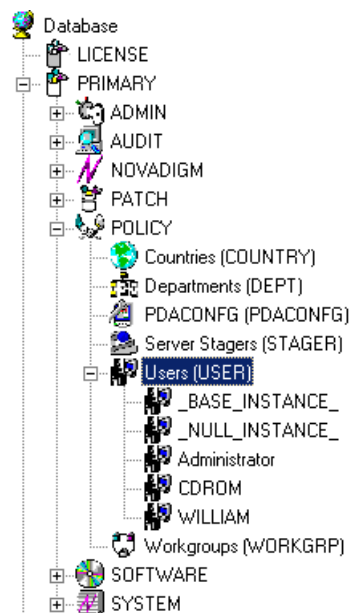
# Building a Distribution Model

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 that is 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 27: PRIMARY.POLICY.Users class.**

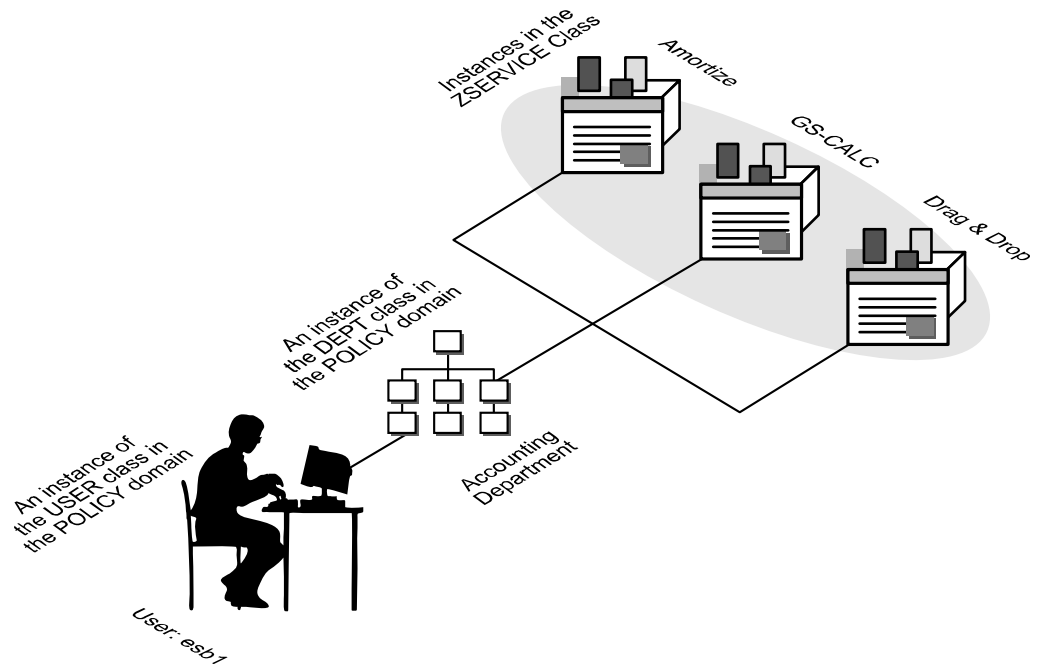
Table 17 below lists the default classes of the POLICY domain delivered with the HP OpenView for Radia Database version 4.1 or above. Refer to the *Database Reference Manual for the HP OpenView Configuration Server Using Radia (Database Reference Manual)* for more information on the POLICY domain classes.

**Table 17: POLICY Domain Classes**

<b>Class</b>	<b>Description</b>
Countries (COUNTRY)	Use for clock synchronizations with the Configuration Server. Do not assign services to this class.
Departments (DEPT)	Use to group subscribers into departments.
Machine Manufacturer (MANUFACT)	Used to assign policy based on the machine manufacturer. Refer to the <i>Installation and Configuration Guide for the HP OpenView Operating System Manager Using Radia (OS Manager Guide)</i> for more information.
Machine Models (MODEL)	Used to assign policy based on the machine model. Refer to the <i>OS Manager Guide</i> for more information.
Machine Roles (ROLE)	Used to assign policy based on the machine role. Refer to the <i>OS Manager Guide</i> for more information.
Machine Subnets (SUBNET)	Used to assign policy based on the machine subnet. Refer to the <i>OS Manager Guide</i> for more information.
Multicast (MULTICAST)	Used to configure client computers to use multicasting. Refer to the <i>Installation and Configuration Guide for the HP OpenView Multicast Server Using Radia (Multicast Server Guide)</i>
Mobile Device Config (MBLCONFIG)	Used to define the parameters for mobile device configuration when using the HP OpenView Mobility Server Using Radia (Mobility Server). Refer to the <i>Mobile Manager Guide for the HP OpenView Management Suite for PDAs using Radia</i> for more information.
PDACONFIG (PDACONFIG)	Used to define the parameters for configuration of cradled-PDAs and to obtain inventory reports of cradled-PDAs. The PDACONFIG class gives the administrator a way of targeting and storing PDA-specific information during resolution.
Server Stagers (STAGER)	Used to define HP OpenView Proxy Servers using Radia (Proxy Servers) or HP OpenView Staging Servers Using Radia (Staging Servers) within your distribution network. Also, use to define

Class	Description
	the storage location on a Staging Server computer.
Users (USER)	Used 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)	Used to group subscribers into functional groups. For example, a project team could 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 28: 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 Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box opens.



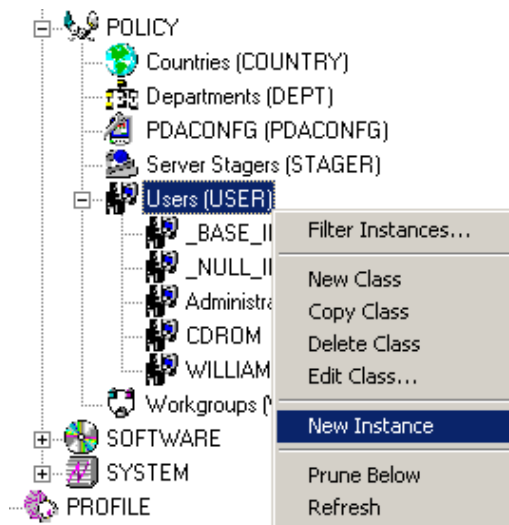
The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might have been changed during installation. Check with your Radia security administrator to obtain your own user ID and password, if necessary.

- 2 If necessary, type a user ID and password, then click **OK**.

The Radia System Explorer splash screen briefly appears.

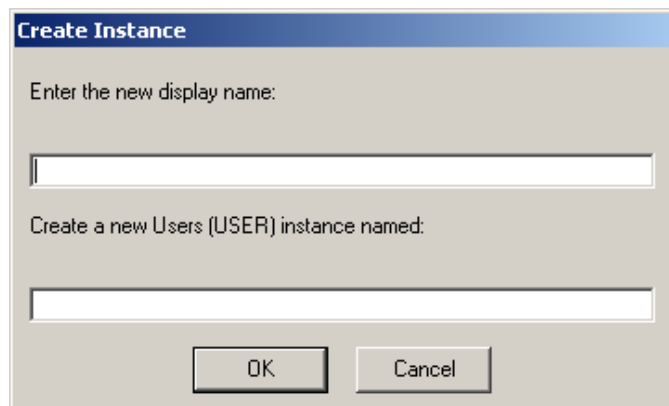
- 3 Double-click **PRIMARY**.
- 4 Double-click **POLICY**.
- 5 Double-click **Users (USER)** to expand the class. A list of all of the USER instances opens.
- 6 Right-click **Users**.

A shortcut menu opens.

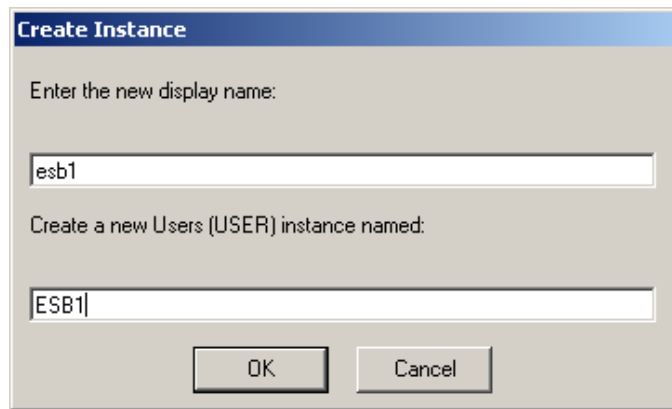


- 7 Select **New Instance** from the shortcut menu.

The Create Instance dialog box opens.

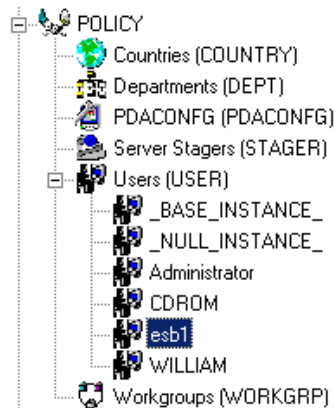


- 8 Enter the display name of the new user. This is the friendly name as it will appear in the tree view. As an example, create a new user instance, ESB1.
- 9 Enter a user's instance name.



10 Click **OK**.

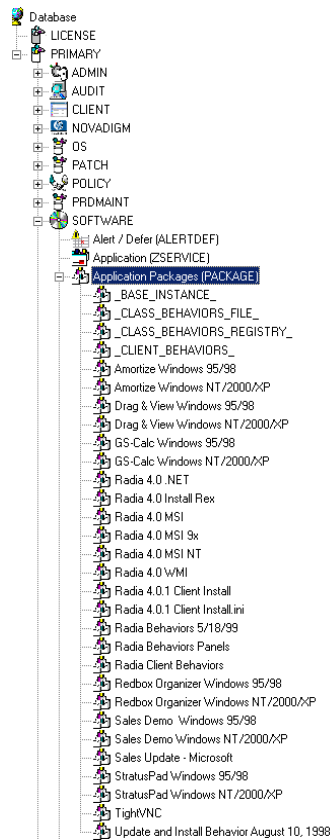
A new **USER** instance, **esb1**, is created.



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

## Software Domain

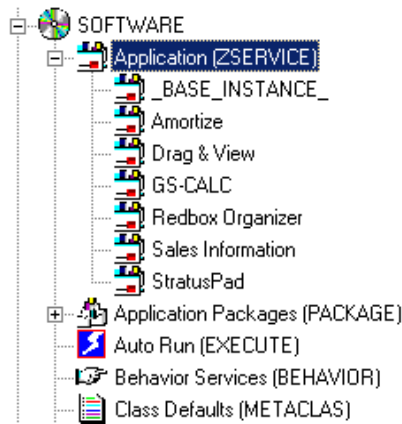
An instance of the **SOFTWARE** domain Application Packages (**PACKAGE**) represents one software application or unit of other content to be managed on client computers. The Radia database is shipped with sample packages as shown in the following example.



**Figure 29: Application Packages (PACKAGE) class.**

Using the HP OpenView Packager Using 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.

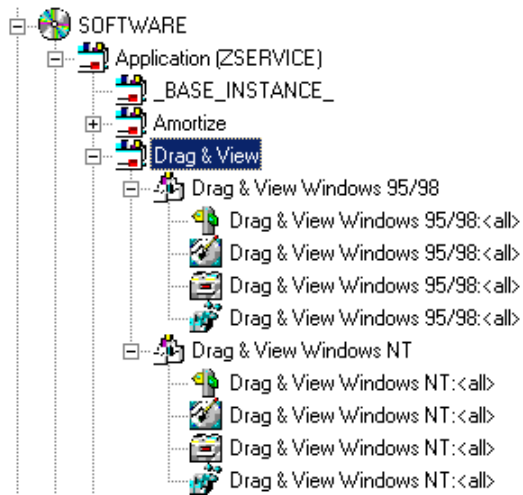


**Figure 30: Sample Application (ZSERVICE) instances shipped from HP.**

For information on packaging and publishing application packages, see the Application Manager Guide or the Software Manager Guide. For information on creating a ZSERVICE, see Running the New Application Wizard beginning on page 136.

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 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 31: 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.

<b>IC</b> _ALWAYS_	Package Includes
<b>IC</b> _ALWAYS_	Package Includes
<b>IC</b> _ALWAYS_	Package Includes
<b>IC</b> _ALWAYS_	Package Includes
<b>IC</b> _ALWAYS_	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>II</b> INCLUDES	Package Includes
<b>IR</b> REQUIRES	Required Package Level
<b>IR</b> REQUIRES	Required Package Level
<b>IR</b> REQUIRES	Required Package Level
<b>IR</b> REQUIRES	Required Package Level
<b>IR</b> REQUIRES	Required Package Level
<b>IR</b> REQUIRES	Required Package Level

**Figure 32: 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 supersede the same components in the required package. For example, if package A requires package B, the components in A supersede 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 might 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 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 Packager and 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 might 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 Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box appears.



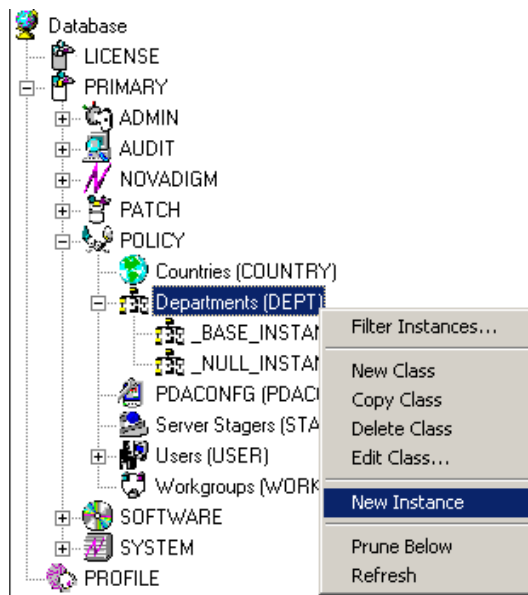
The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might have been changed during installation. Check with your Radia security administrator to obtain your own user ID and password, if necessary.

- 2 If necessary, type a user ID and password, then click **OK**.

The System Explorer splash screen momentarily appears.

- 3 Double-click **PRIMARY**.
- 4 Double-click **POLICY**.
- 5 Double-click **Departments (DEPT)** to expand the class.
- 6 Right click on the **Departments (DEPT)** class.

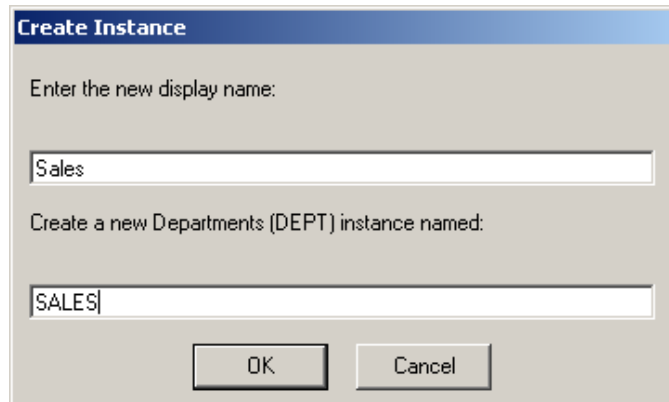
A shortcut menu opens.



- 7 Select **New Instance**.

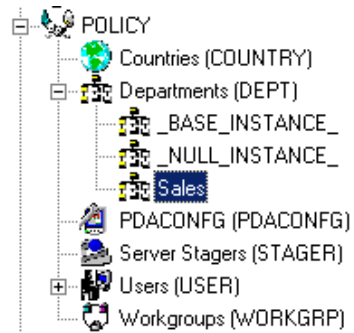
The Create Instance dialog box opens.

- 8 Enter the display and instance names in the appropriate fields. For this example, **Sales**.



- 9 Click **OK**.

The department, **Sales**, is added.



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

## Connection Types

The connecting of Radia database instances to one another can be accomplished in several ways.

### Manual Connections

You can choose to set the connections between instances manually by typing the path of the connection into a Connection attribute. See Table 15 on page 76 for additional information regarding connection types.

- ▶ 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 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:

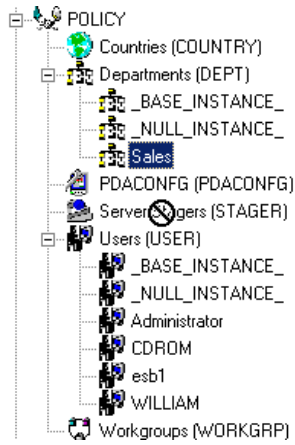
- Typographical error
- Typing in a connection to an instance of a class with which a connection is prohibited.

You can also use drag-and-drop connections between multiple 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.

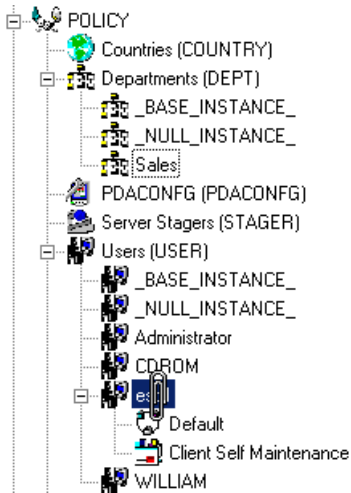
Invoking a drag-and-drop method can be performed rather easily by selecting one instance (clicking and holding down the left mouse button), dragging it to your target instance, and dropping it (releasing the left mouse button) onto your target instance. After you begin the dragging process, you will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



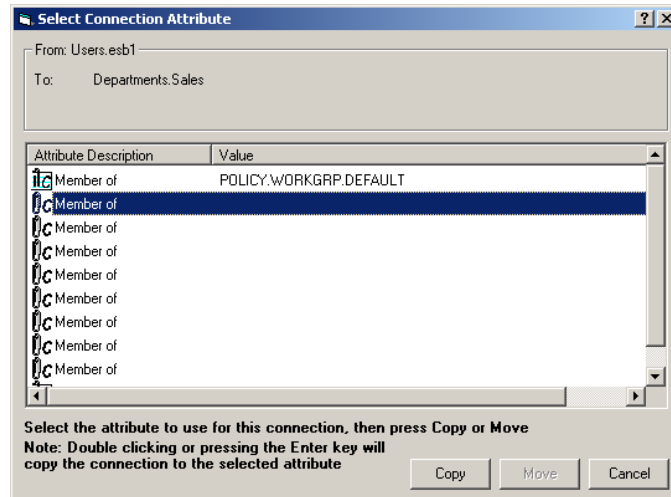
**Figure 33: The circle with a slash indicates that a connection is not allowed.**

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



**Figure 34: The paperclip indicates that a connection is allowed.**

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



**Figure 35: The Select Connection Attribute dialog box showing the next possible connection.**

The System Explorer highlights the first eligible connection attribute in the selected instance, as shown in Figure 35.

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 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.

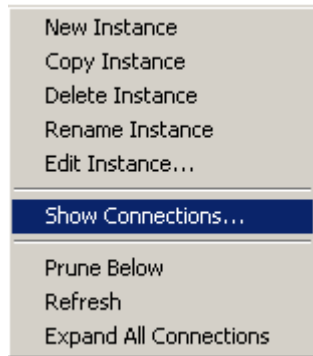
If, instead, you click **Copy**, the connection specification will be left in the instance from which it was dragged, and a copy will be stored in the instance on which it was dropped.

After confirming the connection, the connection is complete.

## Show Connections

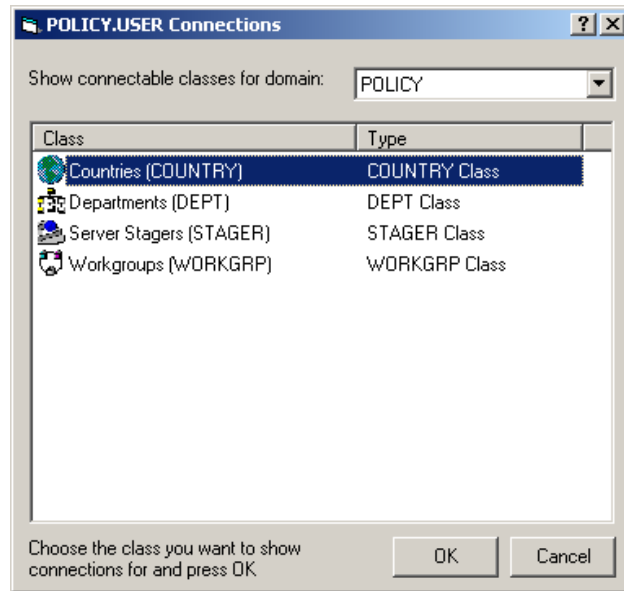
Use the Show Connections dialog box when the instances you wish to connect are not close enough to be displayed in the tree view at the same time.

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



**Figure 36: Select Show Connections to open the Show connectable classes dialog box.**

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

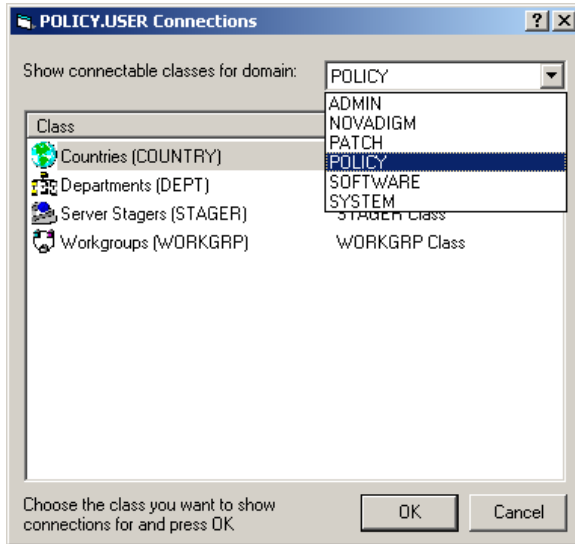


**Figure 37: Show connectable classes dialog box.**

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

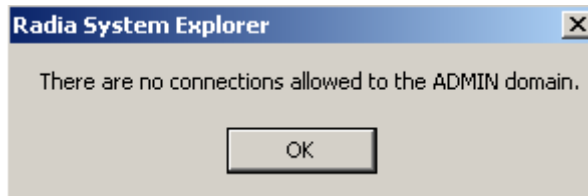
To connect to a class in a different domain, click the drop-down list labeled **Show connectable classes for domain** and a list of domains appears.





**Figure 38: Domain list for connectable classes.**

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



**Figure 39: Warning message for a non-permitted connection.**

Rules for which classes can be connected are contained in the Radia database. See Chapter 7, *Maintaining Connection Rules in the Radia Database* on page 235 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

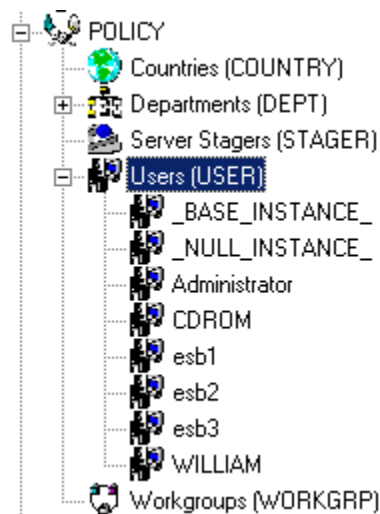
- 1 From the Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box appears.

▶ The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might 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 System Explorer splash screen appears momentarily.

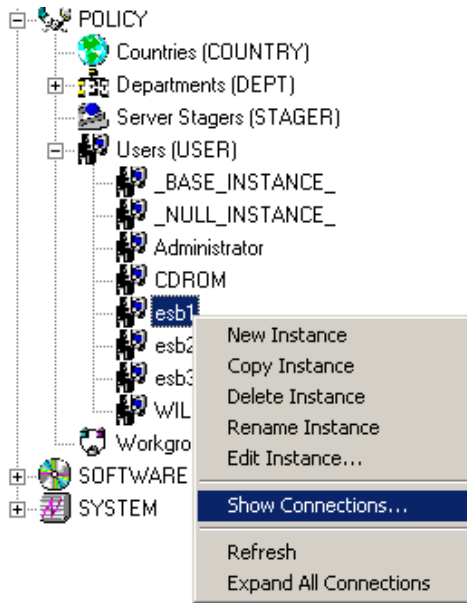
- 3 Double-click **PRIMARY**.
- 4 Expand the **POLICY** domain.
- 5 Double-click **Users (USER)** to expand the class.



**Figure 40: Expanded PRIMARY.POLICY.Users (USER) class.**

- 6 Right-click the **esb1** user instance.

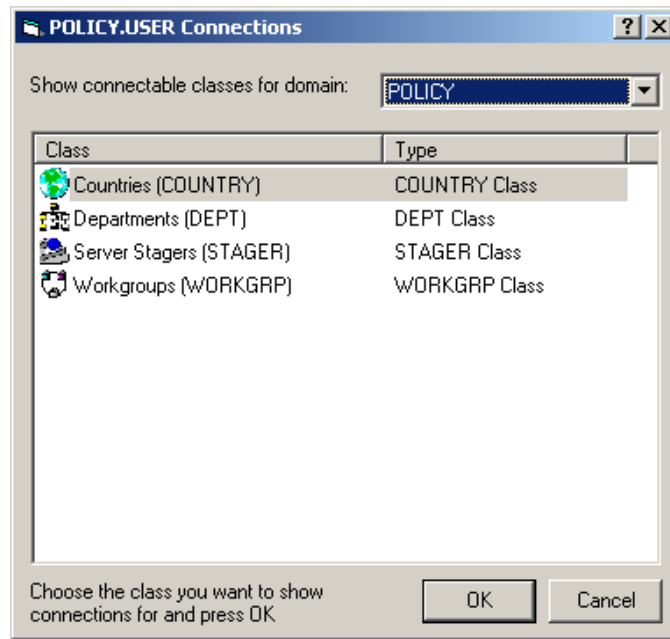
A shortcut menu opens.



**Figure 41: Select Show Connections from the shortcut menu.**

- 7 Select **Show Connections**.

The Show connectable classes dialog box opens.

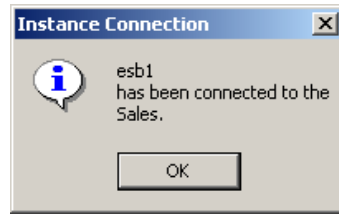


**Figure 42: Show connectable classes for domain dialog box.**

- 8 Select the **Departments (DEPT)** class and click **OK**.

All DEPT class instances are displayed in the list view.





**Figure 44: 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 are 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.




# 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 Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box appears.

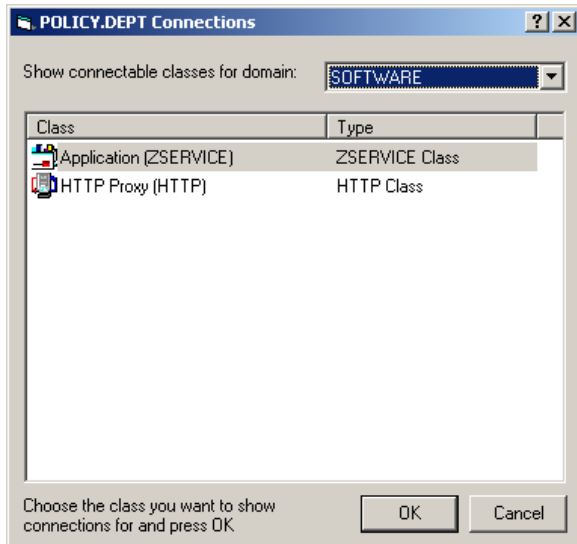
 The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might have been changed during installation. Check with your Radia security administrator to obtain your own user ID and password, if necessary.

- 2 If necessary, type a user ID and password, then click **OK**.

The System Explorer splash screen momentarily appears.

- 3 Double-click **PRIMARY**.
- 4 Expand the **POLICY** domain.
- 5 Double-click **Departments (DEPT)** to expand the class.
- 6 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.
- 7 From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.

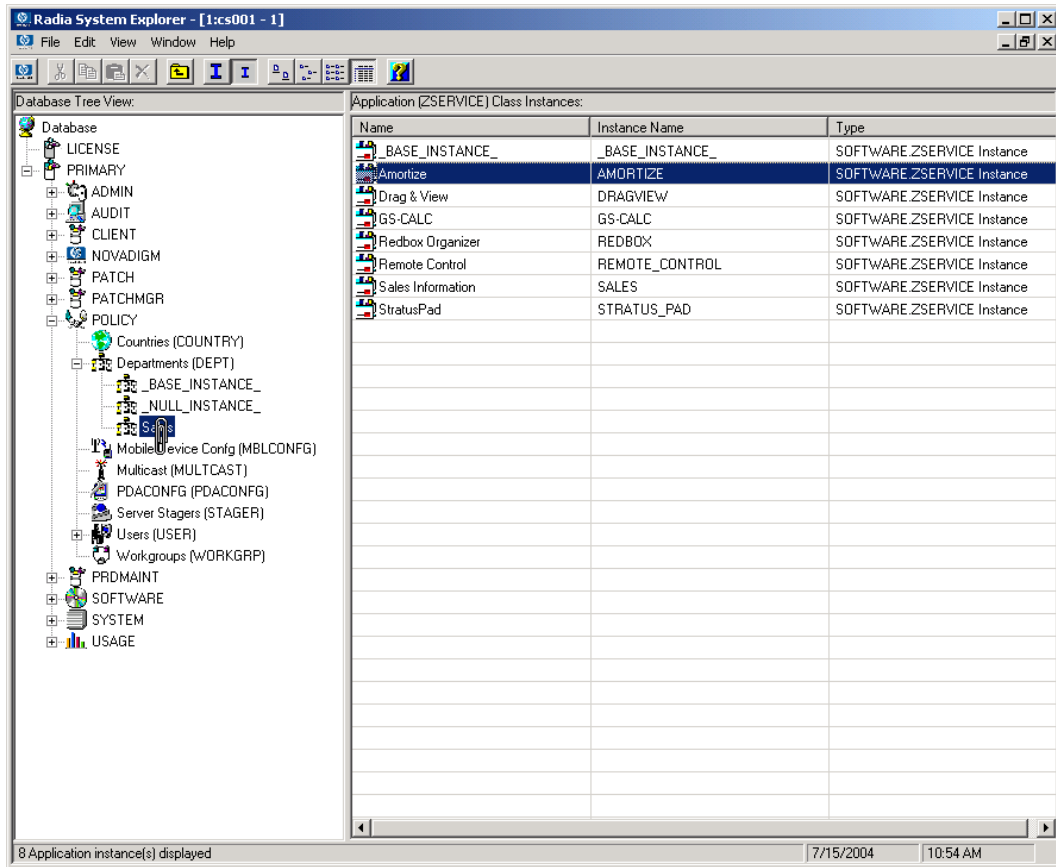




**Figure 46: Select SOFTWARE from the drop-down list.**

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

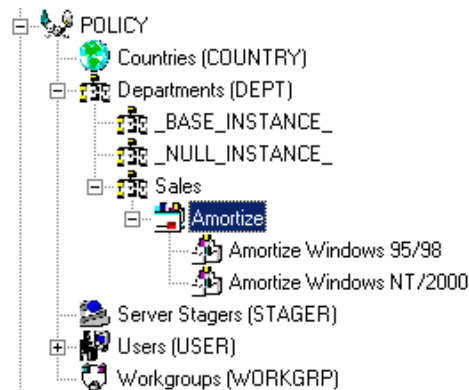




**Figure 48: Select Amortize and drag-and-drop it onto the Sales Departments (DEPT) instance.**

- 10 The Select Connection Attribute dialog box opens.
- 11 Click **Copy** to create the connection from the Sales department to Amortize.
- 12 Click **Yes** to confirm the connection.
- 13 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 49: All users within the Sales Department (DEPT) will receive the Amortize Service.**

## A Word about the Policy Server

The HP OpenView Policy Server Using 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 Configuration Server, determines what services should be distributed and managed for the client that is currently logged on by querying the Policy Server.

The Policy Server plug-in for the 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 Configuration Server.

► The Policy Server and the 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 an enterprise.
- Each client computer managed by Radia is represented by an instance of the POLICY domain, USER class.
- Instances of the SOFTWARE domain, Application Packages (PACKAGE), represent 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 ZSERVICE instance can contain many Application Packages.
- Define the policies that identify which users receive what software.
- 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 Policy Server is a web server used for administrative purposes such as mapping services to users in the directory tree.



# 5 Additional Features of the System Explorer

At the end of this chapter, you will:

- Be familiar with the features of the 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 an application service and have it imported into the database for the HP OpenView Knowledge Base Manager Using Radia (Knowledge Base Manager).

# Features of the System Explorer


This chapter describes these additional features of the System Explorer:

- 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 Knowledge Base Manager database.

For information on publishing content to the Radia database, refer to the *Installation and Configuration Guide for the HP OpenView Application Manager Using 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 Packager or the Packager for Windows Installer.

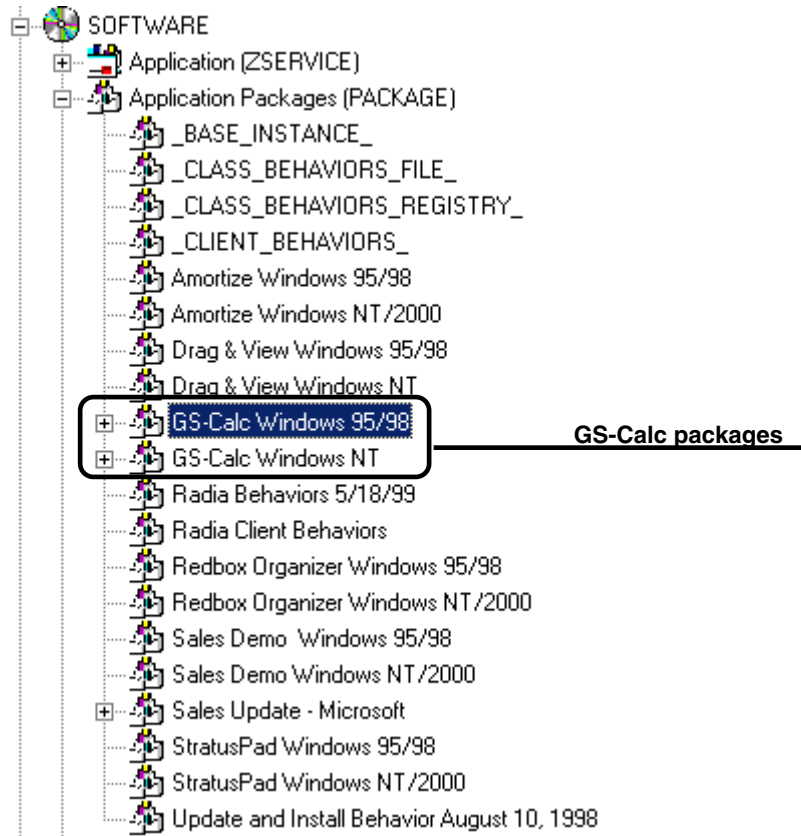
 The GS-CALC package will be used as an example.

To create a GS-CALC service using the New Application Wizard

- 1 Start the System Explorer and navigate to the PACKAGE class of the SOFTWARE domain.

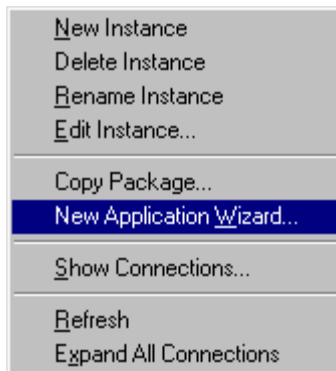


- 2 Locate the two GS-Calc packages in the tree view.
  - One package for **Windows 95/98**; one is **Windows NT**.



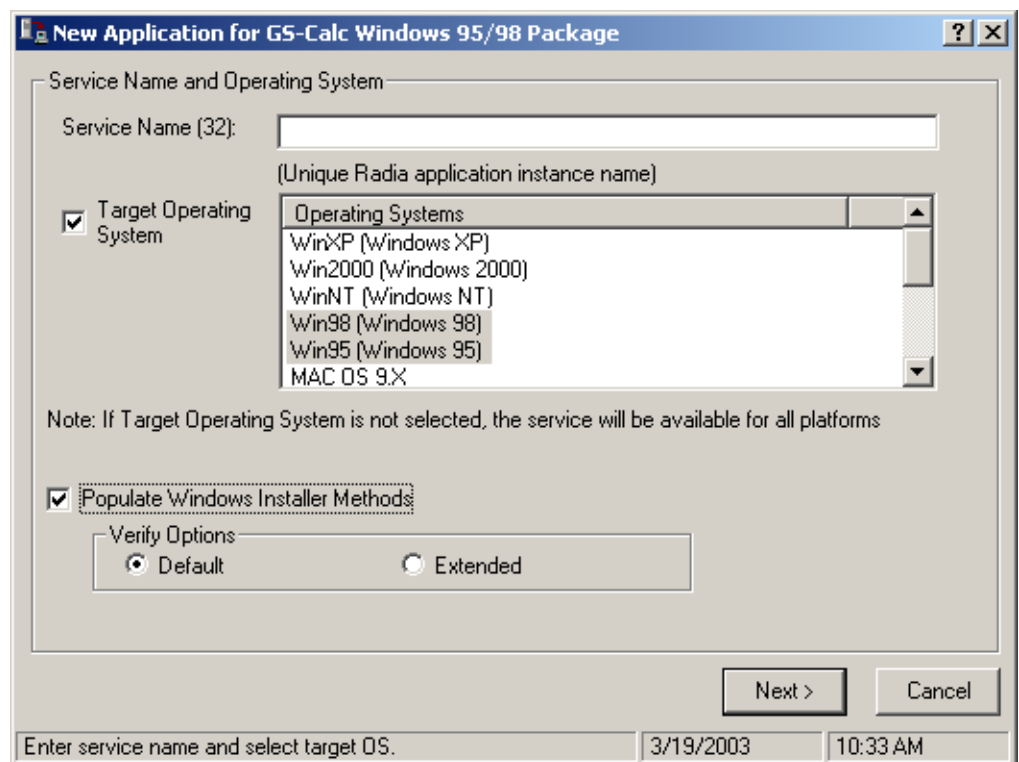
- 3 Right-click the **GS-Calc Windows 95/98** package class instance (highlighted in the above example).

A shortcut menu opens.



- 4 Click **New Application Wizard**.

The New Application Wizard dialog box opens.



► The New Application Wizard dialog box varies for applications created with the Packager for Windows Installer. See *Defining MSI Basic Behaviors on the New Application Wizard* on page 147.

- 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. 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.

► If the package was created with the 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 147.

- 7 If the package was created with the Packager:

Select the **Populate Windows Installer Methods** check box for Windows Installer-enabled applications (e.g., Office 2000). 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**.



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

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

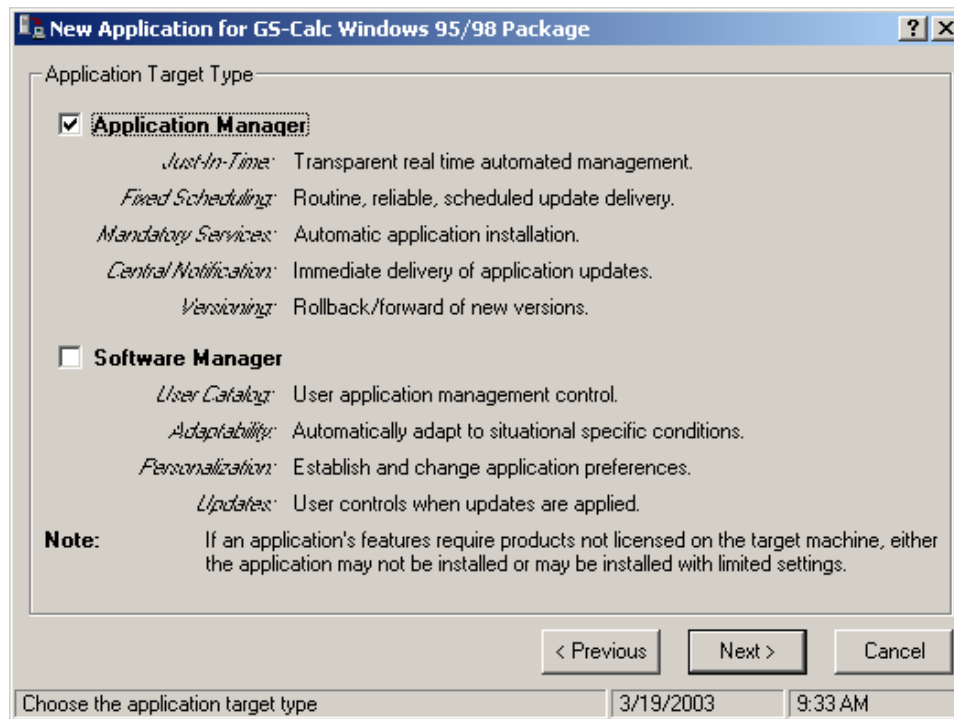
**Table 18: Variables Populated by the New Application Wizard**

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 72.

**8** Click **Next**.

The Application Target Type dialog box opens.



Select one of the two check boxes to indicate which type of support Radia must provide for this application. The differences between the Application Manager and 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.

Fill in the data entry fields as described in Table 19.

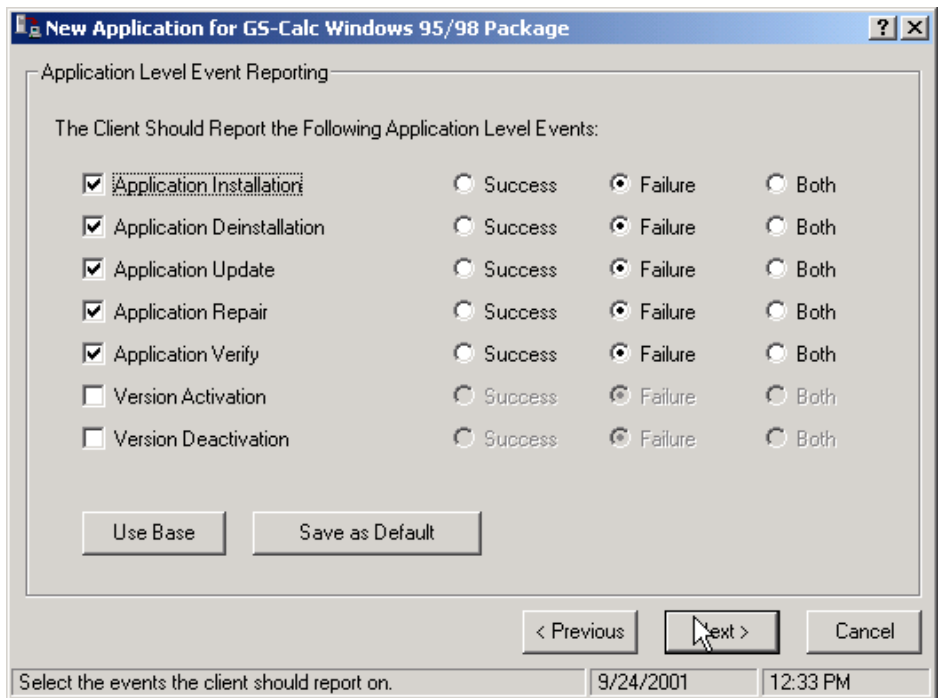
**Table 19: Application Properties Data Entry Fields**

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

<b>Property</b>	<b>Stored in ZSERVICE Attributes</b>	<b>Usage</b>
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.

10 Click **Next**.

The Application Level Event Reporting dialog box opens.



Use this dialog box to indicate which events the Radia client must report to the 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 above requires Radia clients to report failure of application installation, de-installation, update, repair, or verify to the Configuration Server.

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

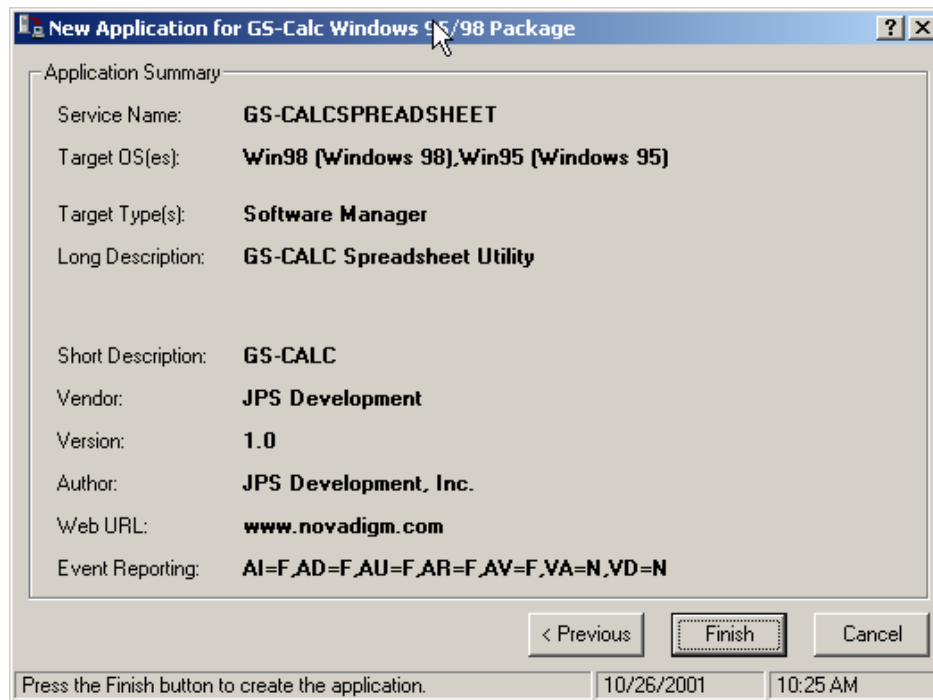
Select **Use Base** if you would like the package to inherit the settings from the base instance of the Application (ZSERVICE) instance.

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.





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 the following table.

**Table 20: Event Reporting Keywords and their Denotations**

<b>Keyword</b>	<b>Reporting Event</b>
AI	Application Installation
AD	Application Deinstallation
AU	Application Update
AR	Application Repair
AV	Application Verification
VA	Version Activation

Keyword	Reporting Event
VD	Version Deactivation

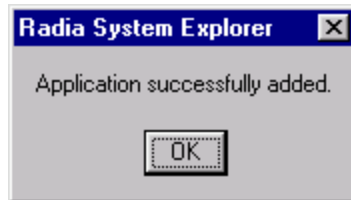
The value associated with each keyword indicates which option button, if any, was selected for the event, as follows:

**Table 21: Values Associated with each Event Reporting Keyword**

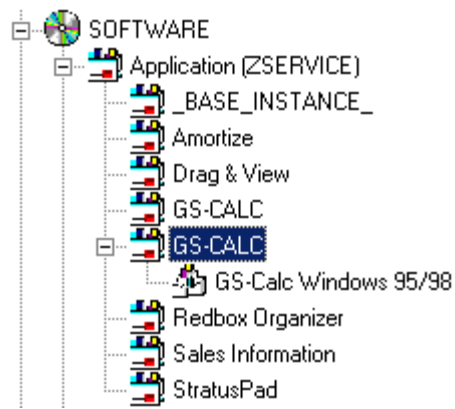
Value	Selected Option Button
S	Success
F	Failure
B	Both (report success and failure)
N	None (no reporting for the event)

- 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.



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

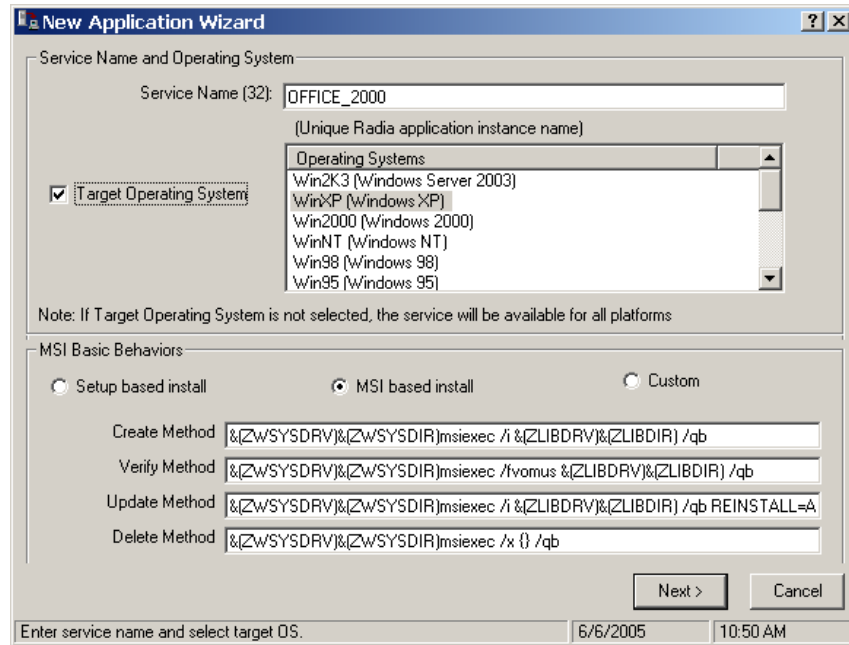


## Defining MSI Basic Behaviors on the New Application Wizard

In contrast to Advanced Package Management, where Radia provides ongoing 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 Packager for Windows Installer. These packages have a connection to the MSIBASIC class in the Radia database.



**Figure 50: 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. This will run the native `setup.exe` for the MSI application. If `setup.exe` supports any command-line switches, such as `/s` for silent or `ADDLOCAL=ALL` for complete installation, you can add them to the **Create Method**. Not all MSI applications support the same command-

line switches. For `setup.exe`, refer to the application's documentation for details.

### **MSI-based install**

The package installs with MSIEEXEC.

### **Custom**

The package uses a custom install method.

- 2 Review and modify, if necessary, the native MSI commands and switches that are to be used for the Create, Verify, Update, and Delete Methods for basic package management.

## The Version Group Editor

The **Version Group Editor** is a feature of System Explorer when the Application Manager is installed. Use it to create and edit instances of the **Version Group** class. Features of the Version Group Editor enable you to create, edit, and delete Version instances of the Version Group and control the Version Group's deployment.

### 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 Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box appears.



The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might 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 System Explorer splash screen briefly appears.

- 3 Double-click **PRIMARY**.
- 4 Expand the **SOFTWARE** domain.
- 5 Right-click **Version Groups (VGROUP)** and select **New Instance** from the shortcut menu.

A **Create Instance** dialog box opens.

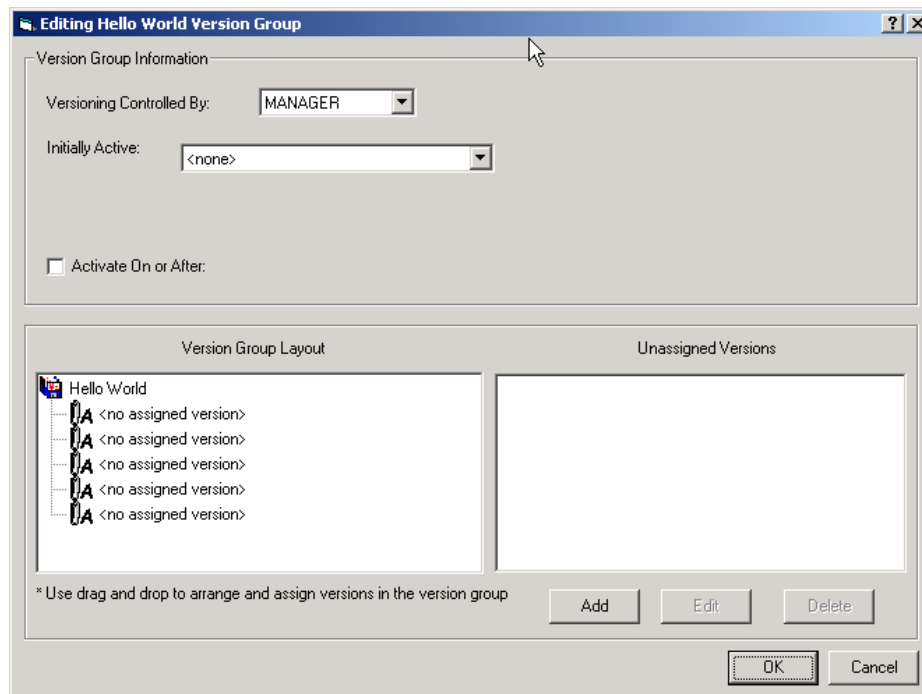


- 6 Type a friendly name for the new Version Group, for example, Hello World.

The friendly name will appear in the tree view of System Explorer.

- 7 Click **OK**.

The Editing Version Group dialog box opens.



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

- 8 Make selections on this dialog box based on the following explanations.

**Versioning Controlled By** drop-down list determines whether the Radia administrator (**Manager**) or end user (**Client**) chooses the version to deploy.

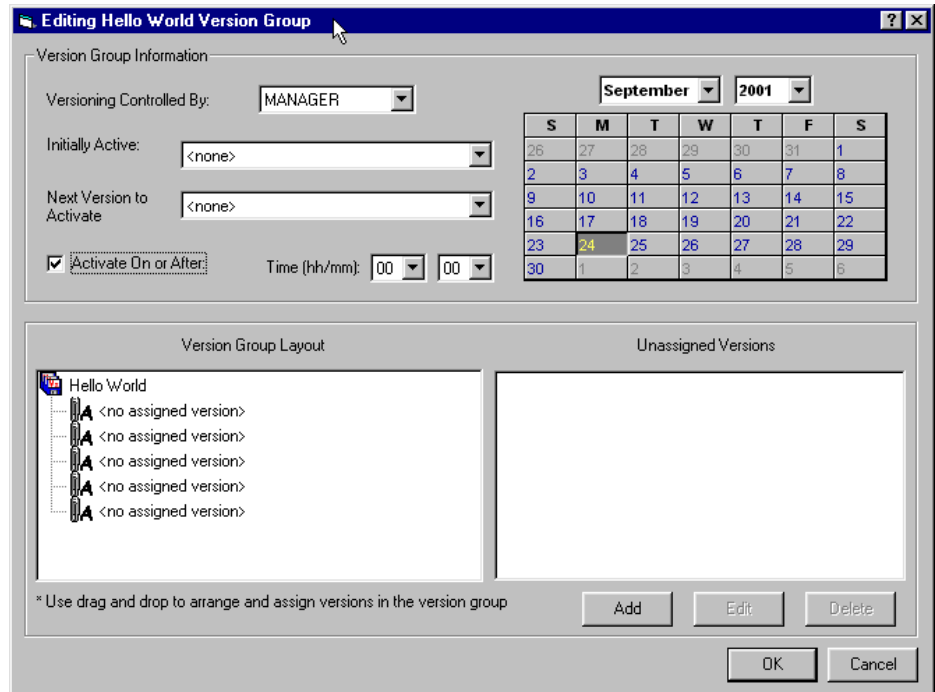
- Choose **MANAGER** to select the Radia Administrator. The Application Manager supports version control by the Radia administrator only; it does not support version control by the end user.
- Choose **CLIENT** to select the end user. This will disable version deployment.

**Initially Active** drop-down list selects which version to activate on the client computer the next time the subscriber's Radia client connects to the 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

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:



Additional features are displayed when **Activate On or After** has been selected.

**Next Version to Activate**

indicate which version of the application to activate if the newly deployed version fails to install and activate successfully.

**Calendar and Time**

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





If using the Version Group Editor to maintain a TIMER instance, consider the following:

- Timers are only created and connected to a VGROUP if versioning is controlled by the 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.

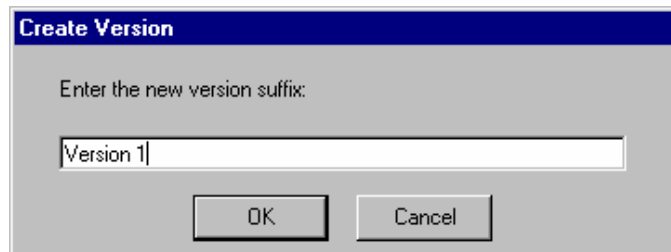
For more information about Timers, refer to the *Application Manager Guide*.

Use the time and calendar controls to indicate the date and time when the designated 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.

- 9 Click **Add** at the bottom of this dialog box.

A **Create Version** dialog box opens.

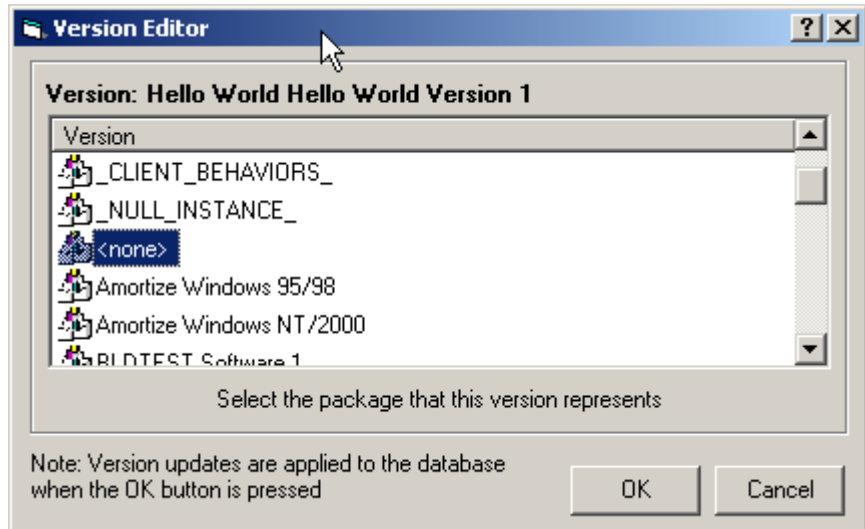


- 10 Enter a suffix for the friendly name for the new Version instance in the text box.

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 System Explorer for this Version instance will be **Hello World Version 1**.

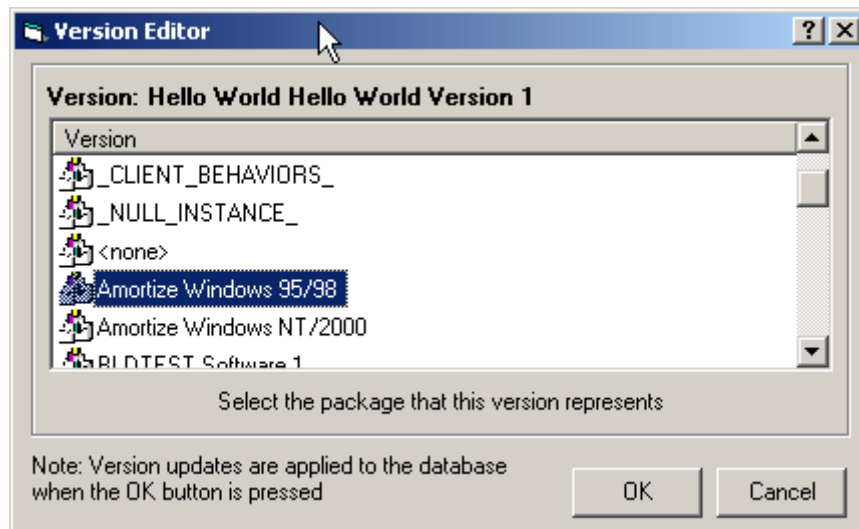
- 11 Click **OK**.

The Version Editor dialog box opens.

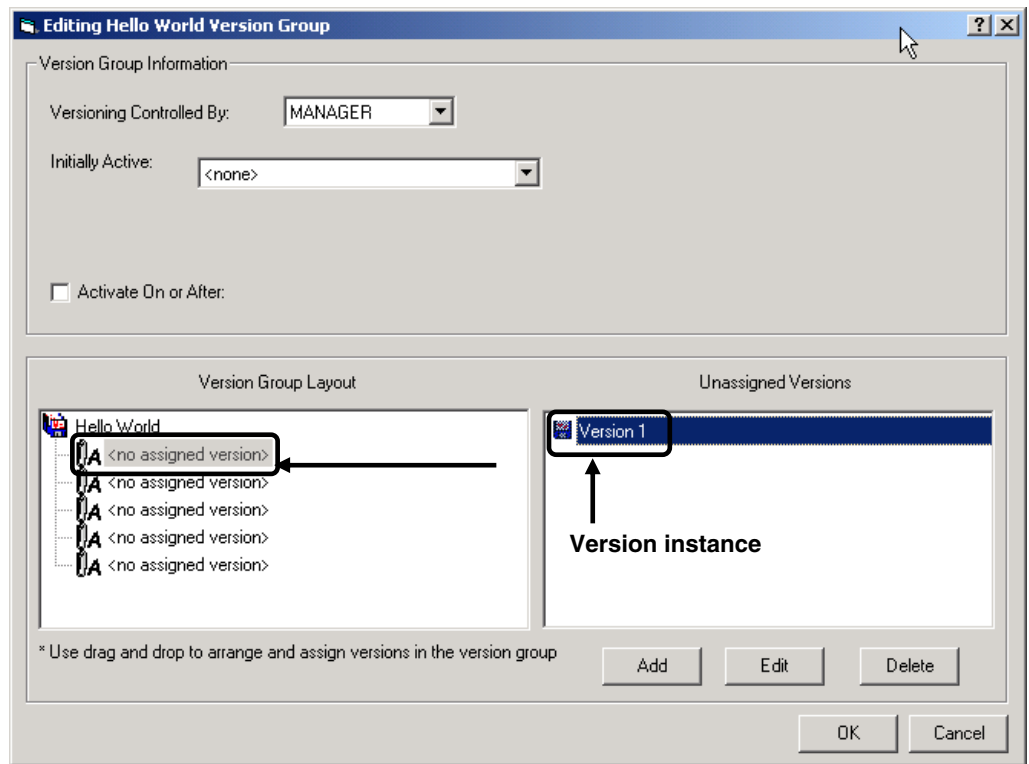


The Version Editor dialog box contains a list of Application Package instances currently stored in the Radia database. Use this dialog box to associate the newly created Version instance with an Application Package instance.

- 12 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**.



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 **Unsigned Versions** list.



- 13 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

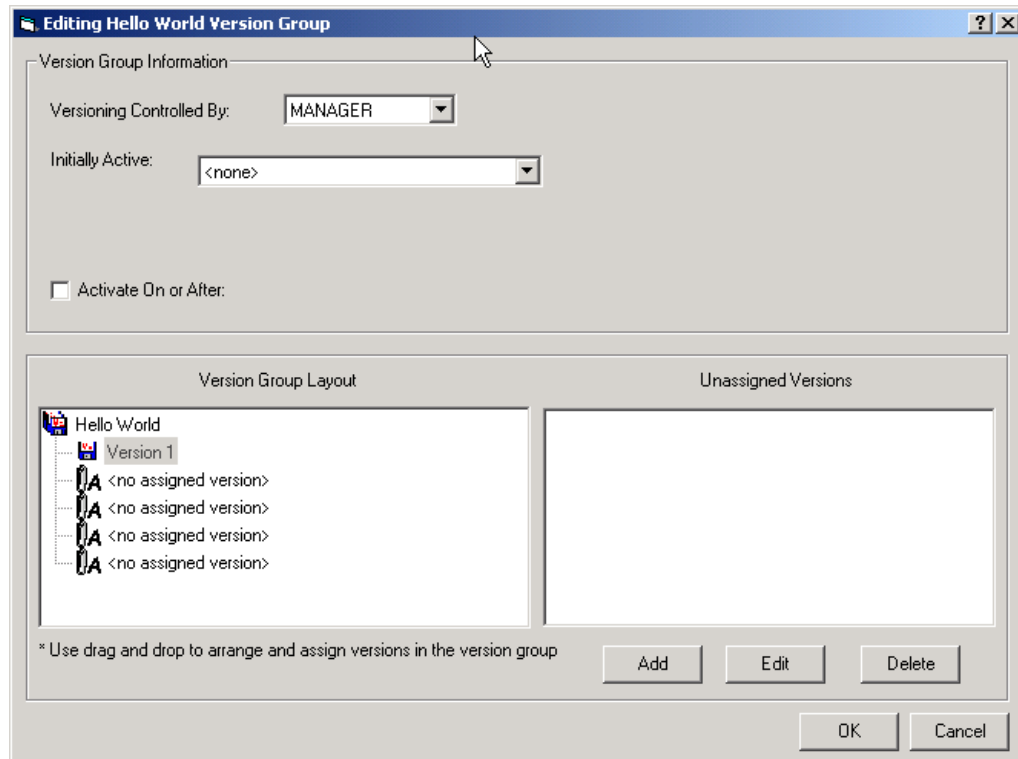
The previous exercise showed how to create a version instance for a package. Once the version is created, you will need to connect it to a Version Group.

### 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.

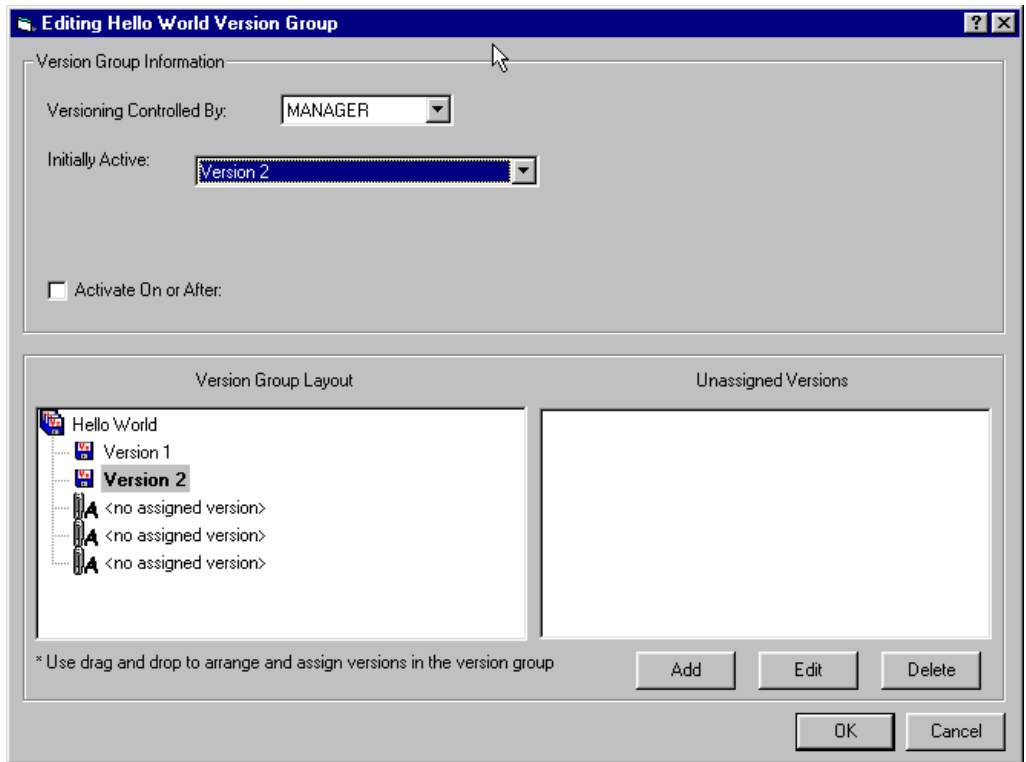
The Editing Version Group window will look similar to:



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

When a Radia client connects to the 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.



If the **Initially Active** control is set to **<none>**, the Radia client connection to the 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 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

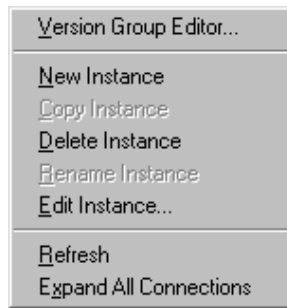
- 1 Highlight the desired Version instance, and click **Delete**. The Version instance will move to the Unassigned Versions list (if it is not already there), and its icon will appear dimmed to indicate that it has been deleted.

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 **Undelete**.

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

### To edit a Version Group

- 1 Right-click the **Version Group** instance in the System Explorer.  
A shortcut menu opens.



- 3 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 HP OpenView Packager Using Radia (Packager), the Radia database, and the System Explorer.



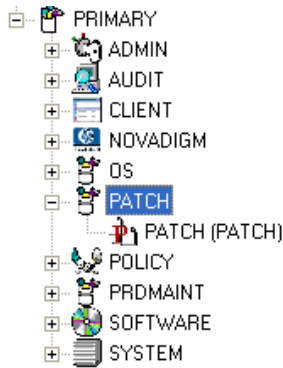
- **Packager**  
The Packager calculates the message-digest algorithm (**MD5**) signature of the file as it is being promoted. The MD5 algorithm can take input and create a 128-bit fingerprint (or message digest) of the input. This information is then stored, in compressed form, in the Configuration Server. For more information about MD5 signature, visit:

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

- **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, refer to the *Configuration Server Guide*.



Prior to deploying patches to users, change the ZOBJPTCH variable from **N** to **Y** in the USER class of the POLICY domain.



**Figure 51: The PATCH class of the PATCH domain in the PRIMARY file.**

The attributes of the PATCH class are described in the following table.

**Table 22: Patch Class Attributes**

<b>Attribute</b>	<b>Description</b>
NAME	The Friendly Name – the value of ZRSCCFIL in the To instance.
STATUS	The status of the patch. Valid values are <b>Queued</b> , <b>Rejected</b> , and <b>Built</b> .
MESSAGE	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 on which the patch was built.
ZRSCTIME	The time the patch was built.
ZRSCSIZE	The size of the patch.
ZCMPFSIZE	The compressed size of the patch.
ZRSCSIG	The content-based signature of the patch type.
ZRSCTYPE	The type of resource. The only valid value is <b>PATCH</b> .
ZRSCCSTA	The status of patch resource. The only valid value is <b>999</b> .
The controls used in patch generation:	
ZCONTRL1	ZSTOPPCT is the percentage of target data when the patch will stop building the path.
ZCONTRL2	ZSTOPMAX is the maximum size of the patch.
ZCONTRL3	ZTIMEOUT is the time allotted to patch creation before aborting the Create. See Table 24 on page 173.
SIGTYPE	Currently, 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.

Attribute	Description
TOFDCI	The original location of the To instance.
TOOID	The object ID of the original To instance.

- **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 Packager. For instructions on how to do this, refer to the *Application Manager Guide*.

- ▶ Patches can be created for components that contain a signature only. 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.
- ▶ Because the Packager now calculates and uses the MD5 signature, and this signature is vital to the patching functionality, there is no backward compatibility to previously promoted components.

This section will provide you with detailed instructions for:

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

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 Configuration Server using the 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.



In some cases, the difference in size between two packages is so minimal that creating a patch will actually increase the size of the file. In this case, the creation of the patch will not be allowed.



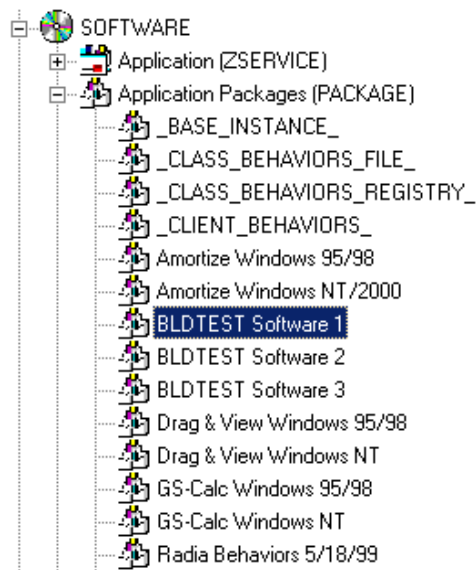
When publishing the files needed for this example, the following naming conventions were used:

- The file `bld1.txt` was originally published as `bld.txt`.
- When the second file was about to be published, it was renamed to `bld.txt`; the original `bld.txt` file was renamed `bld1.txt`.
- When the third file was about to be published, it was renamed `bld.txt`; the second file was renamed `bld2.txt`.

By using the same file name and the same machine during publishing, you are assuring the same CRC needed to correctly create patches.

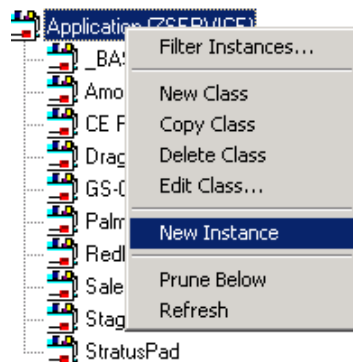
## To configure the Optimize Service feature

- 1 Expand the tree view to the PRIMARY.SOFTWARE.PACKAGE class and locate the BLDTEST packages.



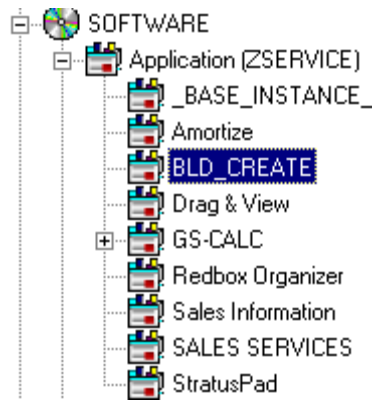
- 2 Right-click on the **Application (ZSERVICE)** class.

A shortcut menu opens.

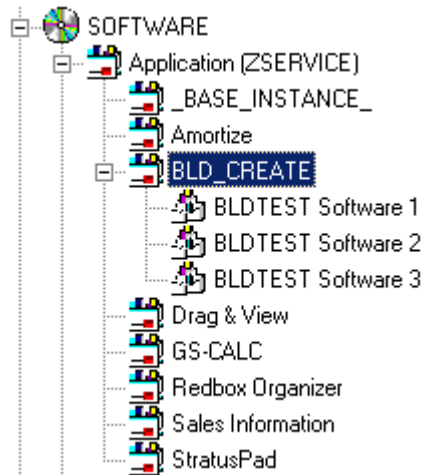


- 3 From the shortcut menu, select **New Instance**.
- 4 Create a new instance named **BLD\_CREATE**, and click **OK**.

The **BLD\_CREATE** service is added to the Radia database.



- 5 Drag-and-drop the BLDTEST1, BLDTEST2, and BLDTEST3 packages to the BLD\_CREATE instance in the Application (ZSERVICE) class.



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

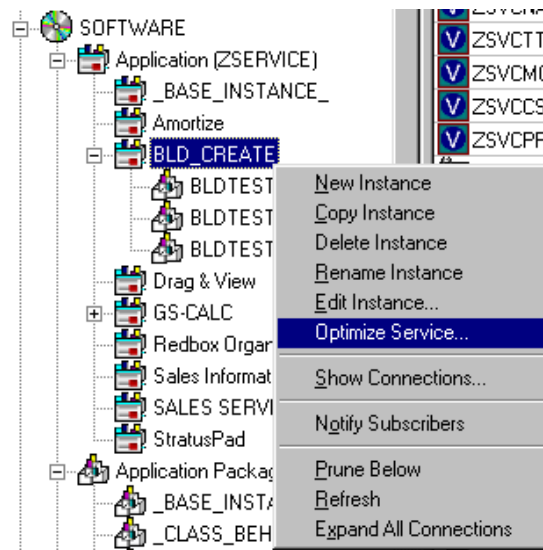
Just as you create a plan or policy for users, you might want to develop a plan for creating, editing, and deleting patches. Some important things to remember when creating patches:

- It is important to devise a means by which to document the packages that will create your patches. When building different patches 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.

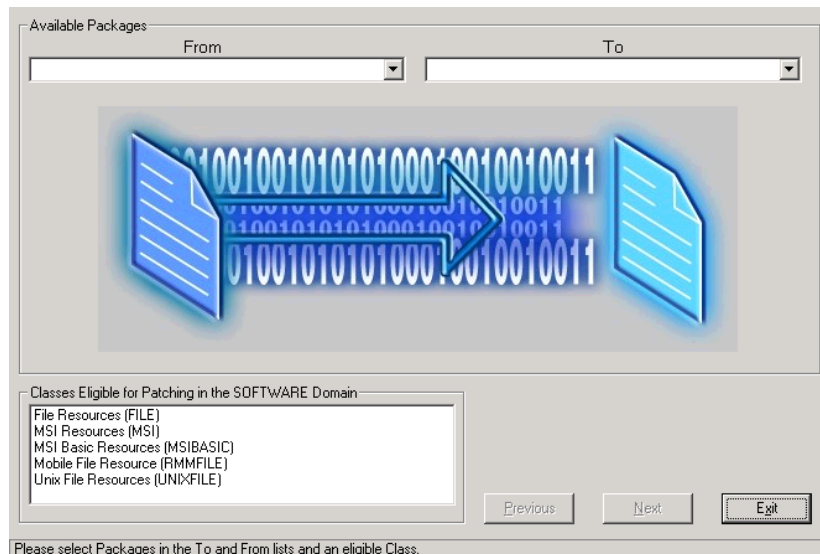
- Only those packages that contain an MD5 signature are eligible for patching.
  - Patching is different from *versioning*. If a patch fails to deploy properly, there is no previous version option.
- 6 Right-click on the Application (ZSERVICE) class **BLD\_CREATE**.

A shortcut menu opens.



- 7 Select **Optimize Service**.

The **Service Optimization** wizard opens.



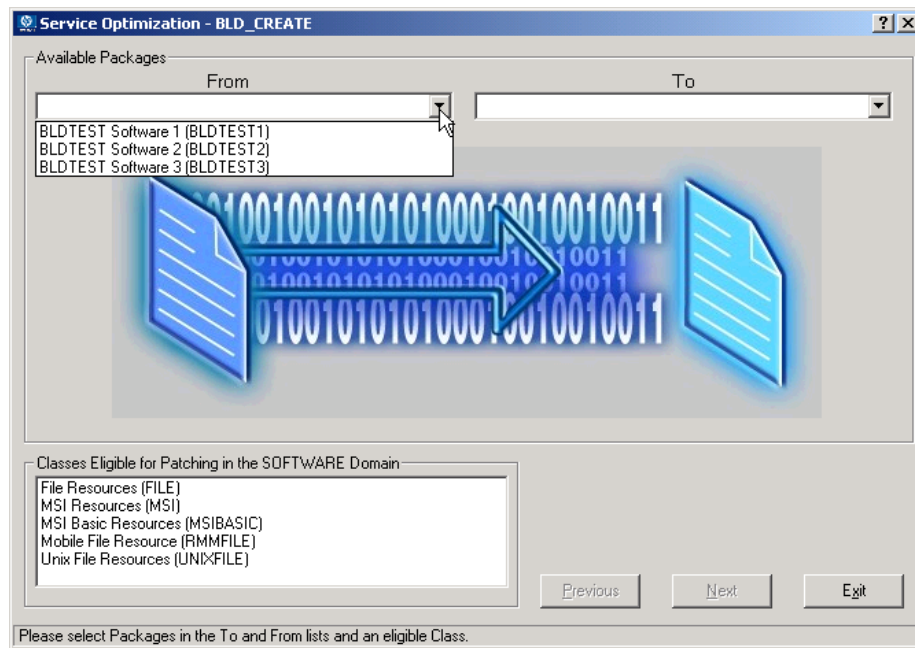
There are three areas in this first screen of the **Service Optimization** wizard:

- **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 (ZSERICE) 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.

- 8 Click the **Available Packages From** drop-down arrow.

The packages that are available for the service you selected in step 6 (on page 167), BLD\_CREATE, are listed.

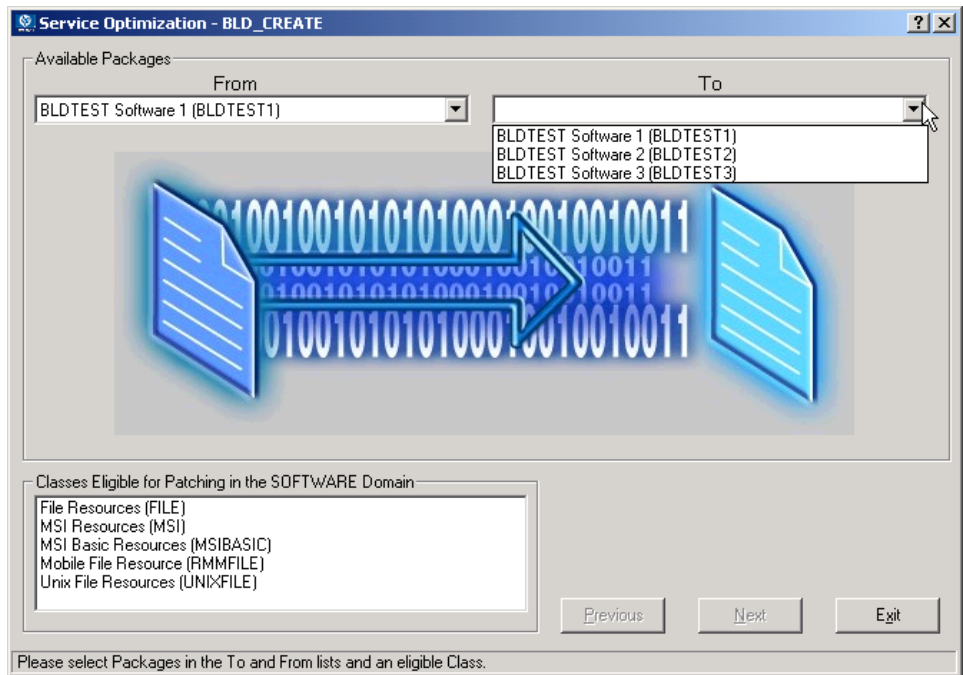




- 9 From the drop-down list select **BLDTEST Software 1 (PACKAGE 1)**.

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

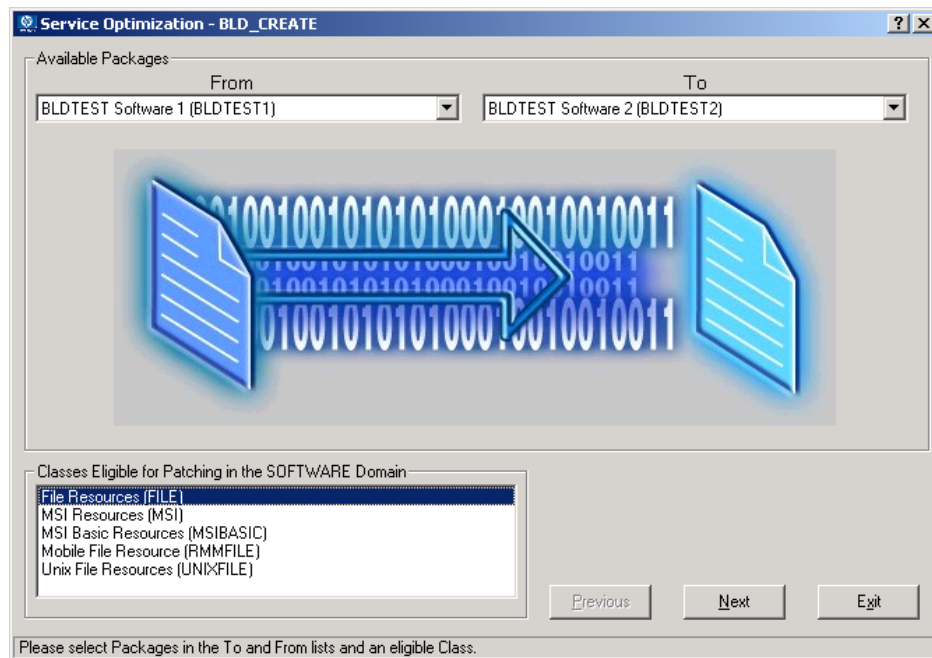
- 10 From the **Available Packages To** drop-down list select **BLDTEST Software 2 (SOFTWARE 2)**.



11 From **Classes Eligible for Patching in the SOFTWARE Domain** select the appropriate class for patching in the SOFTWARE domain. Choose the class in which the file resources for the package are located.

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

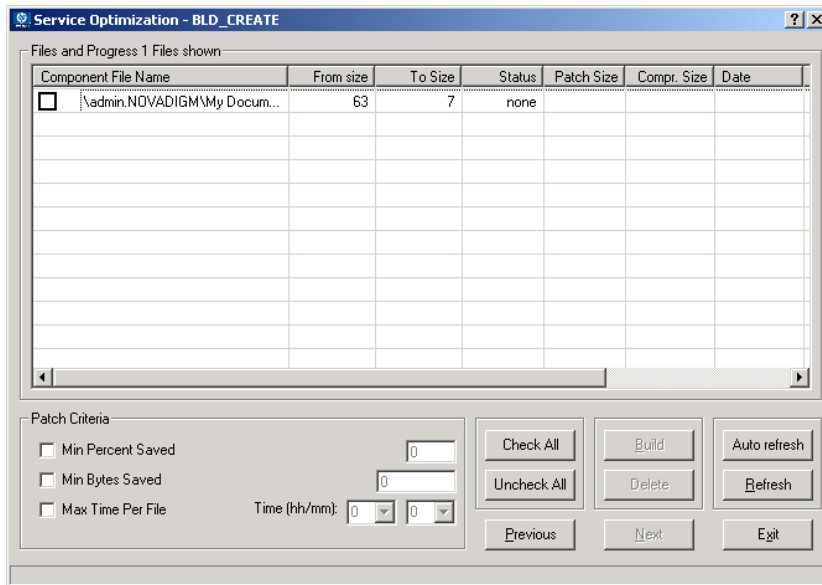
In this example, the only class that is available to be patched is **File Resource (FILE)**.



- ▶ To determine which classes are available for patching within your package, expand your Application (ZSERVICE) instance to the attribute level. Examine your instances to determine which resources are available for patching.

12 Select **File Resource (FILE)** and click **Next** to continue.

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



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

The options in the **Files and Progress** section of this dialog box are described in Table 23. The options in the **Patch Criteria** area are described in Table 24 on page 173.

**Table 23: Files and Progress Information**

Column	Description
Component File Name	The name of the patch that is being created.
From Size	The original resource-package size.
To Size	The original target-resource size.
Status	The status of the package. The valid values are <b>none</b> , <b>queued</b> , <b>rejected</b> , <b>active</b> , and <b>built</b> .
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 <b>From Size</b> .
Date	The date on which the package was published and promoted.

Column	Description
Size Limit	This information is pulled from the <b>Patch Criteria, Min Percent Saved</b> 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 <b>Patch Criteria, Min Bytes Saved</b> 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 <b>Patch Criteria, Max Time Per File</b> 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, and time elements by configuring the **Patch Criteria** options. You can control the creation of patches by setting criteria that will apply to all files whose check boxes are currently marked in the file list.

Suppose that the number of files in the list is large and you don't want 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** options to limit the building of patches to those files likely to produce a benefit.

Table 24 details the options of the **Patch Criteria** area. The default for these three options is an empty (unselected) check box.

**Table 24: Patch Criteria Information**

File Option	Description
Min Percent Saved	Specify the minimum percent of the To component to save. Otherwise, the patch will not be built.
Min Bytes Saved	Specify the minimum number of bytes that the patch must save. Otherwise, the patch will not be built.
Max Time per File	Specify the maximum processing time for the patch. Use the drop-down list to indicate the maximum time allowable for patch building. The time can be set in minutes, or hours and minutes.

Once a component file has been selected, the buttons used for patch building are enabled.

To apply a criterion, select its check box. The adjacent data entry fields 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.

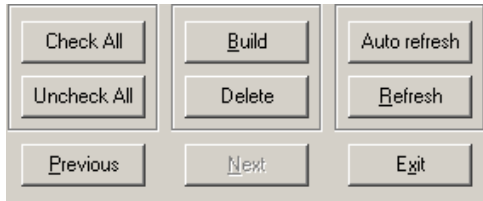
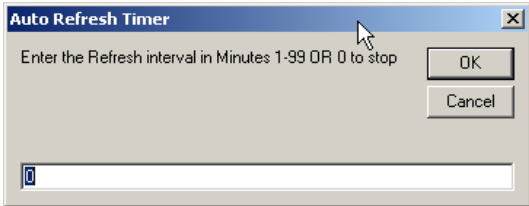


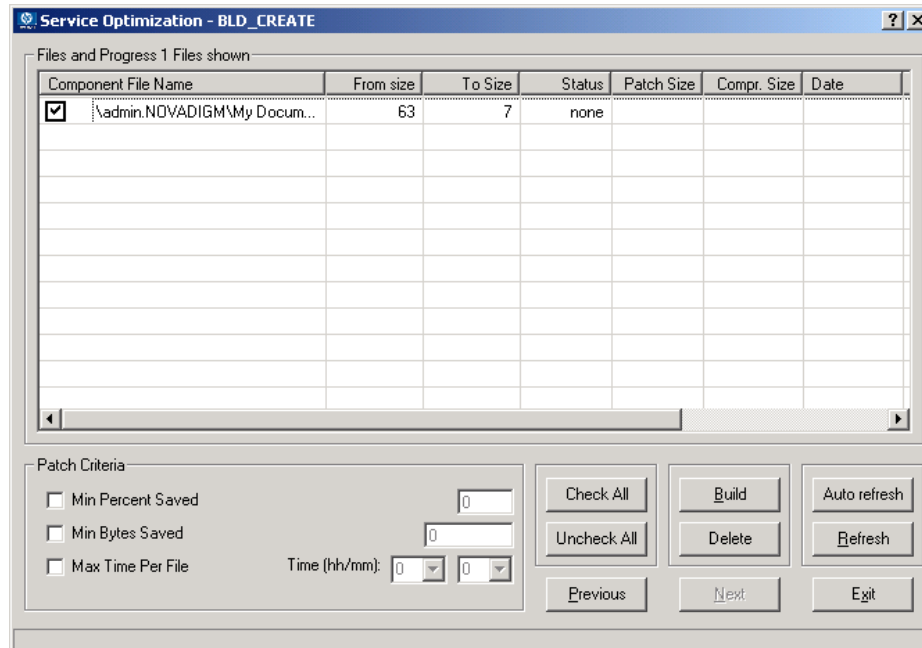
Table 25 below explains the functions of the Patch Criteria buttons.

**Table 25: Patch Criteria Button Functions**

Button	Action
Check All	Selects all of the files that are listed in the <b>Component File Name</b> column.
Build	Click here to begin building patches.
Auto Refresh	<p>Selecting <b>Auto refresh</b> opens the <b>Auto Refresh Timer</b> dialog box.</p>  <p>When using Auto Refresh, enter a refresh interval between 1 and 99 minutes. You can cancel the auto refresh by entering 0.</p>
Uncheck All	De-selects all files that are selected in the <b>Component File Name</b> column.
Delete	Deletes a selected component file.
Refresh	Refreshes build status.

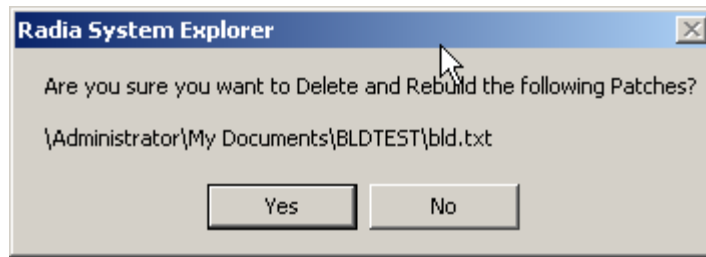
Button	Action
Previous	Return to the previous screen.
Next	When enabled, click <b>Next</b> to proceed to the next screen.
Exit	Click <b>Exit</b> to exit the Optimization Wizard.

13 Select the check box next to the files to be patched.



14 Click **Build**.

System Explorer will prompt you to confirm the new patch build.



15 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 Configuration Server.

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

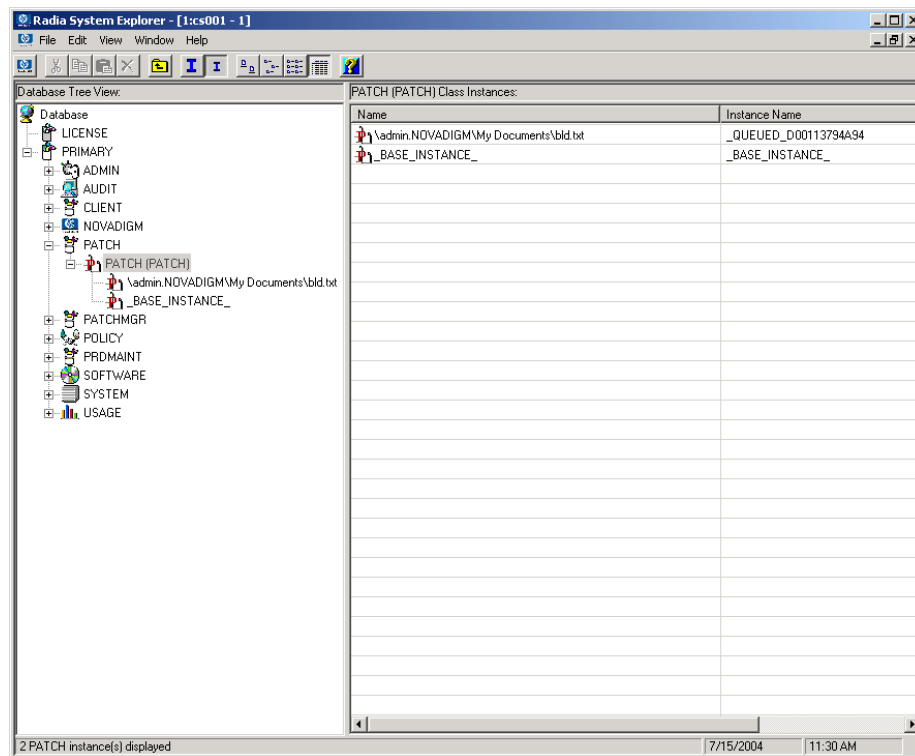
17 Click **Exit** to exit the Service Optimization wizard.

or

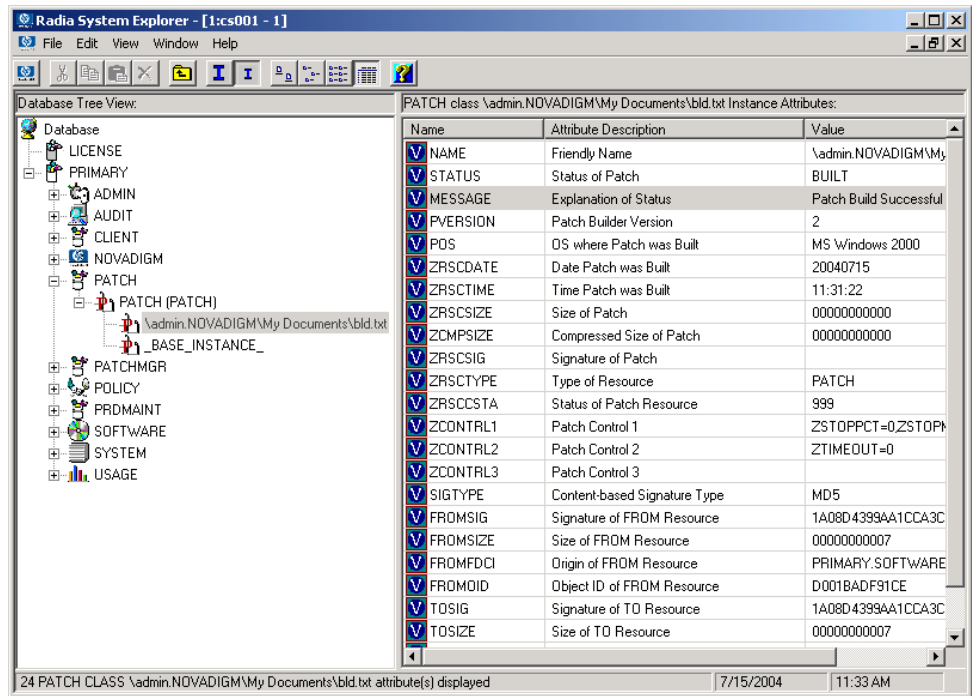
Click **Previous** to return to the **Available Packages** dialog box to build additional patches.

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





Examine the PATCH.PATCH and verify that the patch instances are built. Examine rejected patches to determine why they were rejected. See the MESSAGE variable to determine the current status of requested patches.



## Radia Service Groups

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

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

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

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

- ▶ Remember to target actions, such as notifies or timer updates, to the master service. Any action applied to the master service is applied to all sub-services.

## Component Enhancements

- **The Software Manager**  
Each service within the Radia **Service List** will display only the master service.
- **The Application Manager**  
The RADSKMAN command contains the logic to process multiple service requests. It 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 service for a service group, change or confirm the following settings:

- Configuration Server (*versions 4.5 and later*), in the `edmpprof` file, verify that `ALLOW_DUPLICATE_INSTANCES` in the `MGR_OBJECT_RESOLUTION` section is set to **No**.
- Configuration Server (*versions prior to 4.5*), in the `edmpprof` file, verify that in the `MGR_OBJECT_RESOLUTION` section:
  - `ALLOW_CIRCULAR_REFERENCE` is set to **Yes** and,
  - `ALLOW_DUPLICATE_INSTANCES` is set to **No**.
- Radia database

To configure a set of services (a grouping in which services depend on one another for installation), add a variable to the `_BASE_INSTANCE_` of the Application (ZSERVICE) class. This is a variable called ZSVCGRP. If you

would like to add this variable to the ZSERVICE class, use the following table to help you set the variable.

For information on editing a class, see *Editing a Class* beginning on page 72.

**Table 26: ZSVCGRP Variable Values**

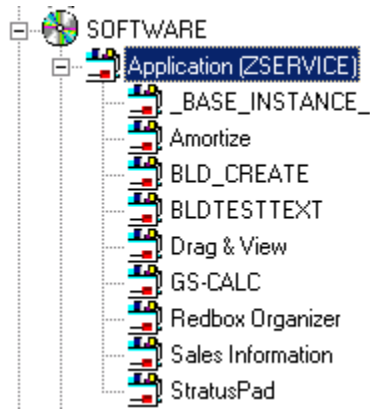
Value	Behavior
Y	Specify <b>Y</b> (Yes) for the Configuration Server to process the service groups as if they are unrelated applications or applications that are not dependent on one another for installation.  If this variable is not present, this is the default.
D	Specify <b>D</b> (Dependent) for the Configuration Server to process the services under the service group as though they are dependent on one another for installation. If one sub-service fails, the remaining services will not be installed.
N	Specify <b>N</b> (No) for no service groups.

## Creating a Master Service for a Service Group

Once the changes have been made to the Configuration Server `edmprof` file you can begin creating a **master service** instance for a service group.

To do this, you will use the System Explorer to create a new Application (ZSERVICE) instance for the master 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 service. You can add service members to the master service at any time.

For our example, we will be creating a master service instance called **Sales Services**. This master service will contain the **Amortize** and **GS-Calc** services that have been provided in the sample Radia database.

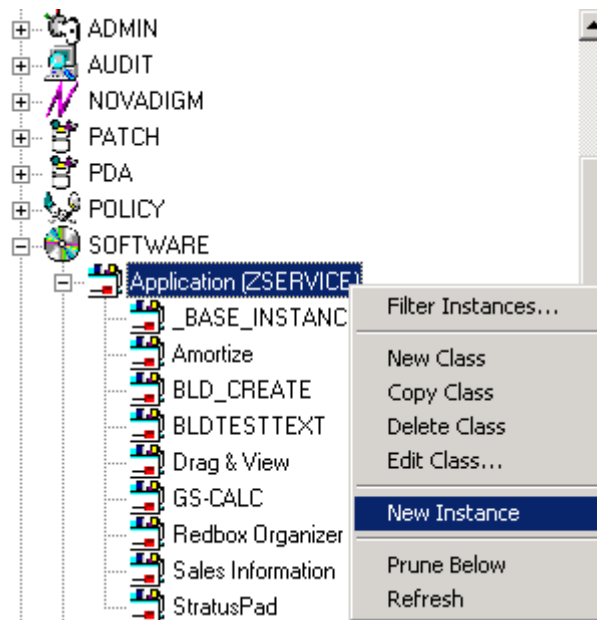


**Figure 52: Sample services in the ZSERVICE class.**

### To create a Master Service for a Service Group

- 1 Open the 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)**.

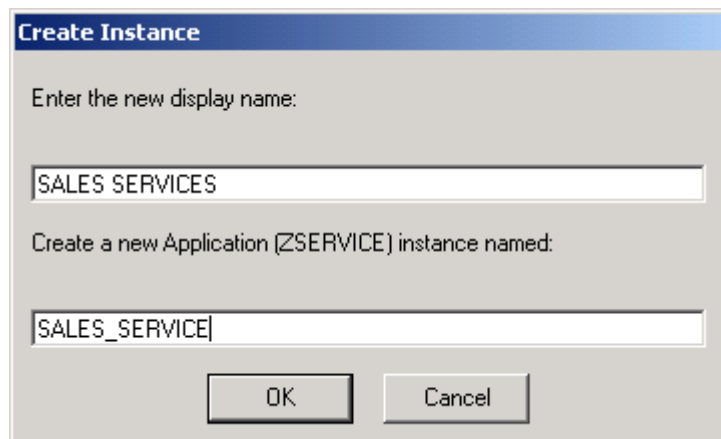
A shortcut menu opens.



- 4 Select **New Instance** from the shortcut menu.

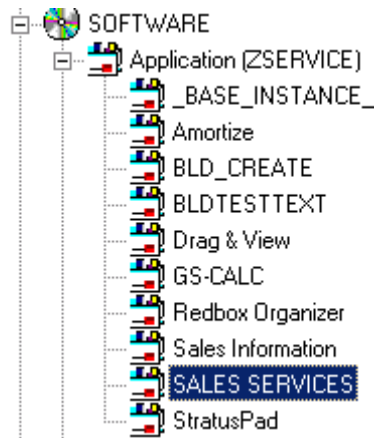
The **Create Instance** dialog box opens.

- 5 Enter the name of the new ZSERVICE instance (Sales Services) that you are creating.



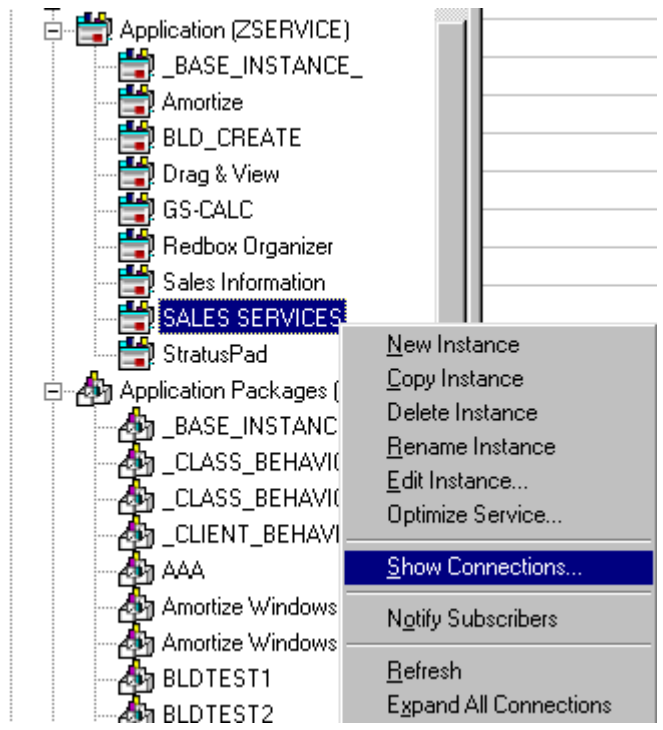
- 6 Click **OK** to create the instance.

The new instance appears in the Application (ZSERVICE) class.



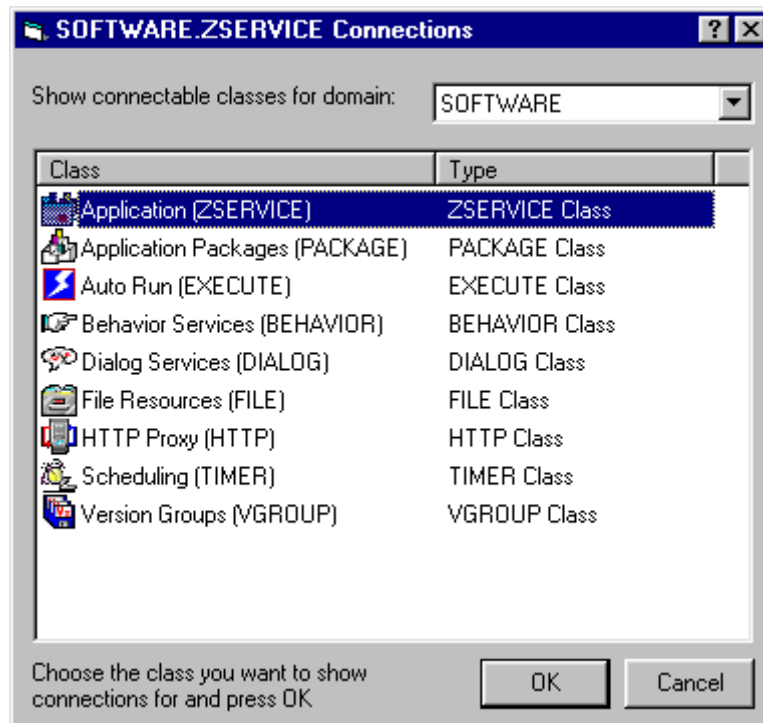
You are ready to add services to the master service.

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



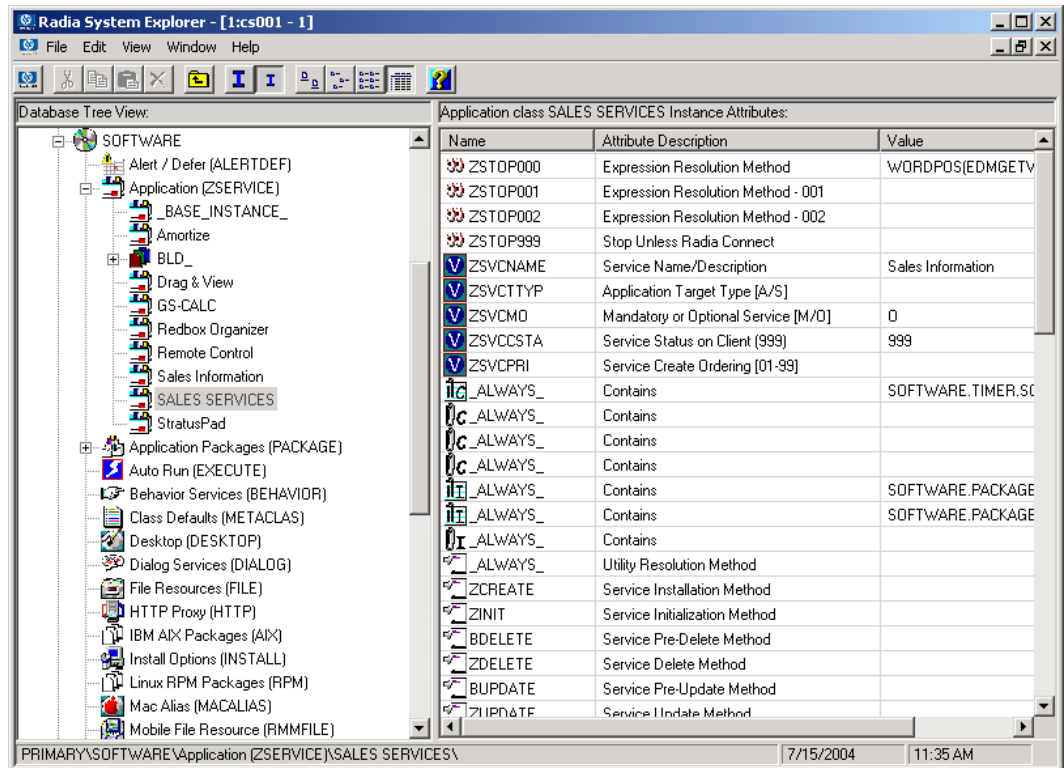
The **SOFTWARE.ZSERVICE Connections** dialog box opens.





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

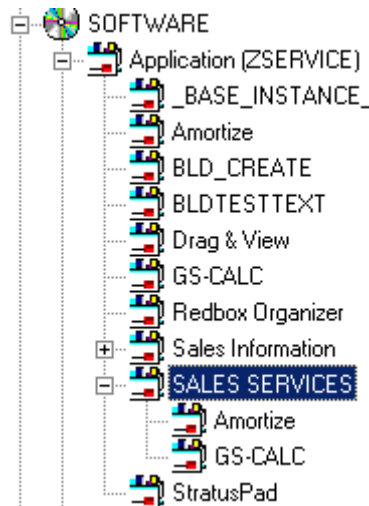
A list of the ZSERVICE class instances is displayed on the list view of the screen.



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

The master service instance for this service group is now complete.

➤ The services grouped into a master service will install in visual order; that is, from top-down according to the order in which they are connected to the master service instance in the database. Member services in a master service group do not honor the settings in ZSVCPRI.



This master service is now ready to be deployed to your users.

## Radia Notify

Radia Notify enables the Radia administrator to cause one or more client computers to connect to the Configuration Server and to update or remove a specified application. It can also send e-mail notification to client computers, based on the applications that Radia manages for them.

► For detailed information on Radia Notify requirements, refer to the Deploying Applications chapter of the *Application Manager Guide*.

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 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, 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 Configuration Server must be properly configured for Notify. For information on setting the appropriate values for the Configuration Server Settings file, `edmprof`, refer to the Engineering Note, *OV-ENKB01129: Configuring the Radia Configuration Server for E-mail Notification* 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 **Options** dialog box. You can customize the UID and STARTDIR values when notifying subscribers to install an application. For details, see the Notify Tab on page 50.

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

- Drag a POLICY class instance (USER, DEPT, and WORKGRP) 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.



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, 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 Open the System Explorer and navigate to the **PRIMARY.SOFTWARE.ZSERVICE** class.

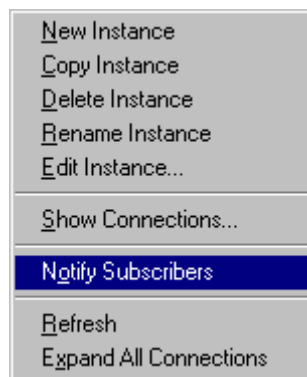
- 2 Expand the Application (ZSERVICE) class.
- 3 Navigate to and expand the PRIMARY.POLICY class.
- 4 Initiate a notify action using a drag-and-drop operation or a menu-based selection.

#### **Drag-and-drop Notify**

To initiate a drag-and-drop Notify, select a POLICY instance (workgroup, department, or individual user), drag-and-drop it onto an Application instance.

#### **Menu-based Notify**

To initiate a menu-based Notify, right-click on an Application instance. The following menu opens.

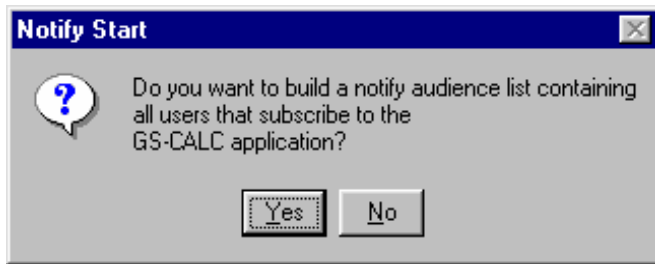


- 5 Select **Notify Subscribers**.

## 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.

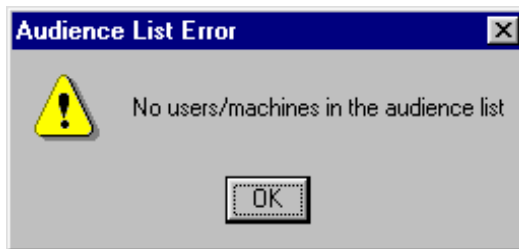


1 Select **Yes** to continue.

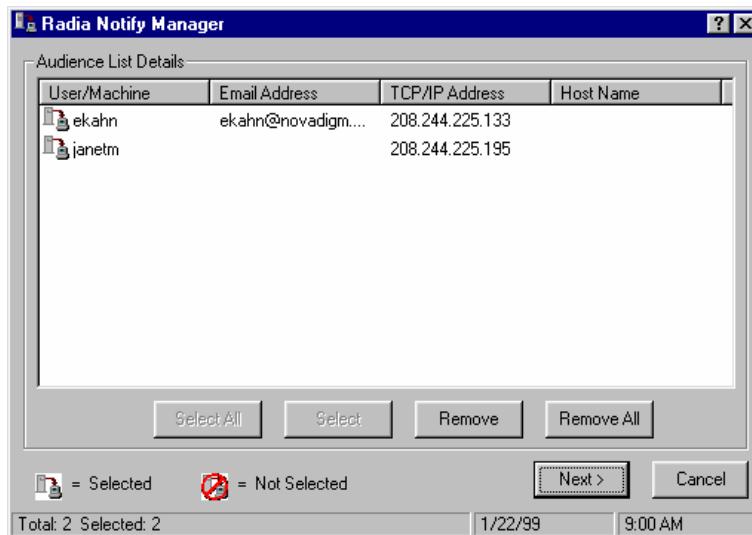
or

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.



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



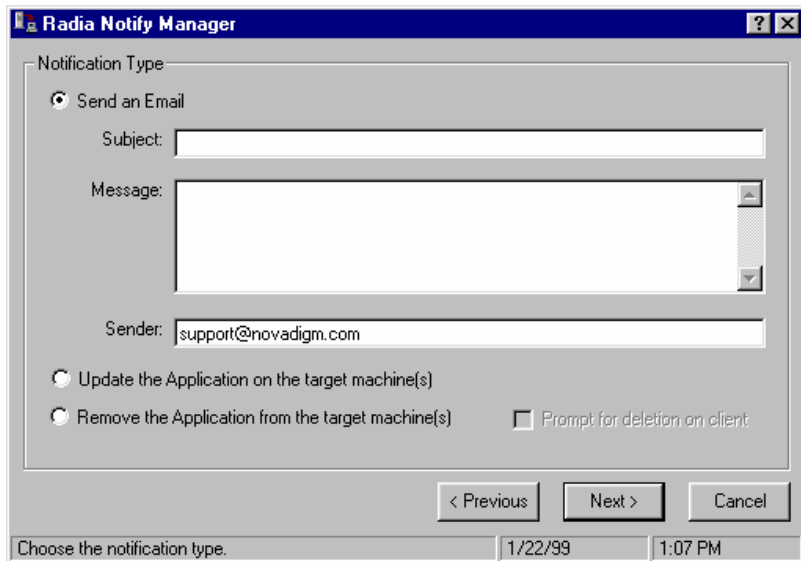
- 2 To notify the entire audience set, click **Select All**. This is the default when the audience list is first displayed.

To generate a tailored Notify audience, highlight the row associated with a subscriber and choose either **Select** or **Remove**. Before using **Select**, 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.

- 3 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.



- **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.



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

- **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)**

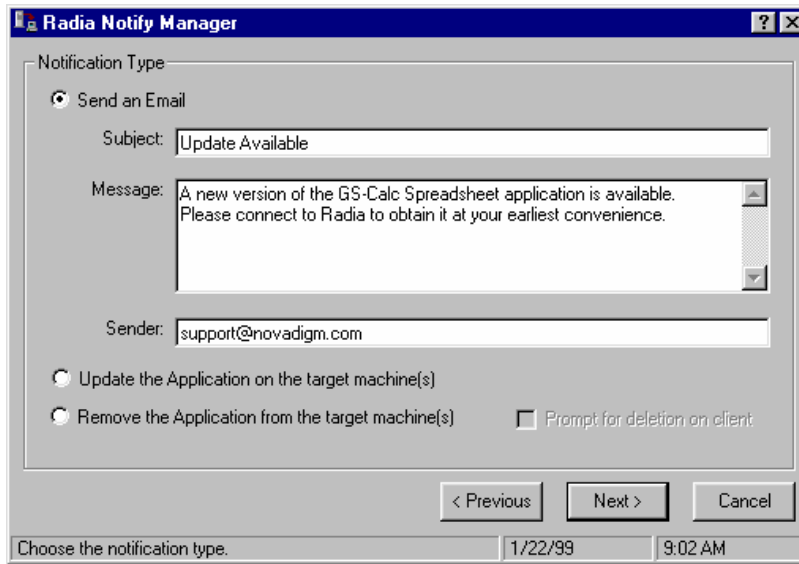
Use this option to remove an application from the specified desktops.

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

## Send an E-mail Using Notify

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





**Figure 53: 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)**.

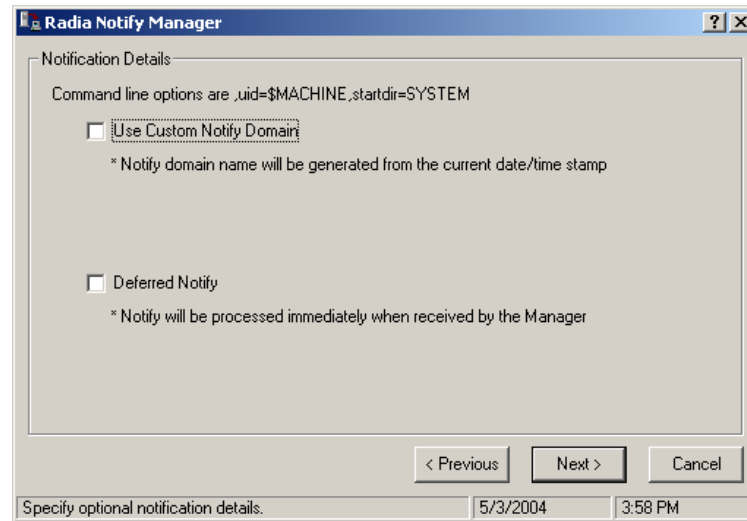
When you choose this option, the **Prompt for deletion on client** check box becomes enabled. Normally, Notify will remove the application from the selected client computers in the audience list without requesting permission from the end user. This allows unattended application removal from client computers.

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 54:** 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.

- To reset the Notify command values for UID or STARTDIR, cancel out of this dialog box, set the **Notify** options for UID and STARTDIR using the **Options** menu, and then repeat the Notify Subscribers selections. See Notify Tab on page 50 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 can choose to leave the **Notification Details** check boxes empty. This will cause the (following) default settings to be used.

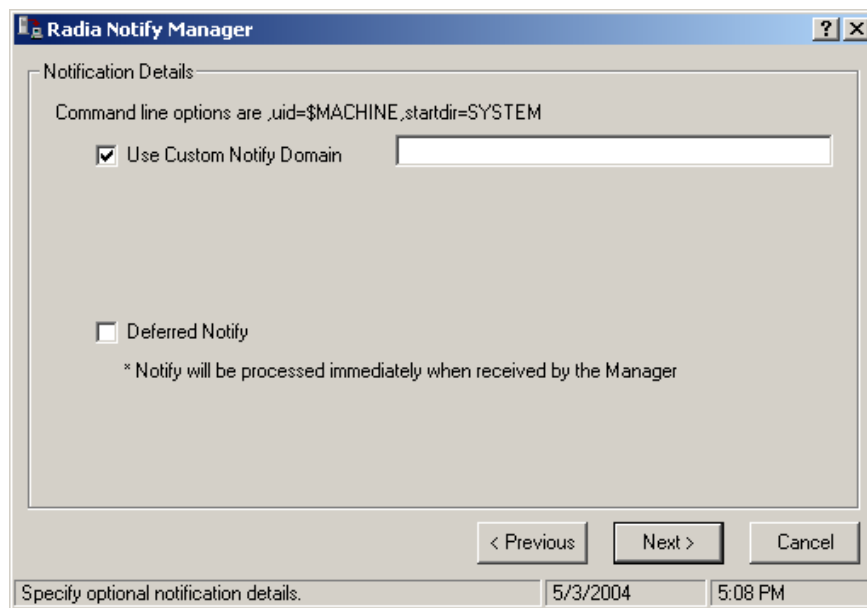
- The Notify domain name will be generated from the current date/time stamp.
- The deferred notify date/time will be processed immediately when received by the Configuration Server.

### To specify a domain name

- 1 Select the **Use Custom Notify Domain** check box.

A text box opens.

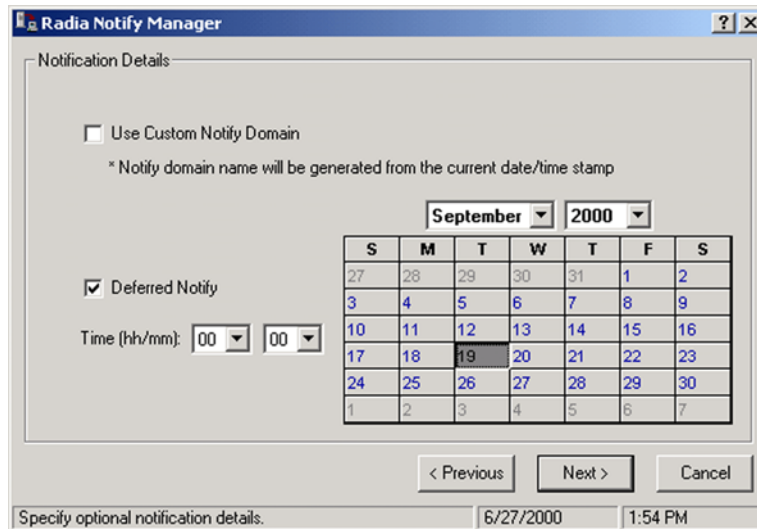
- 2 Type a domain name.



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 199 for additional information.

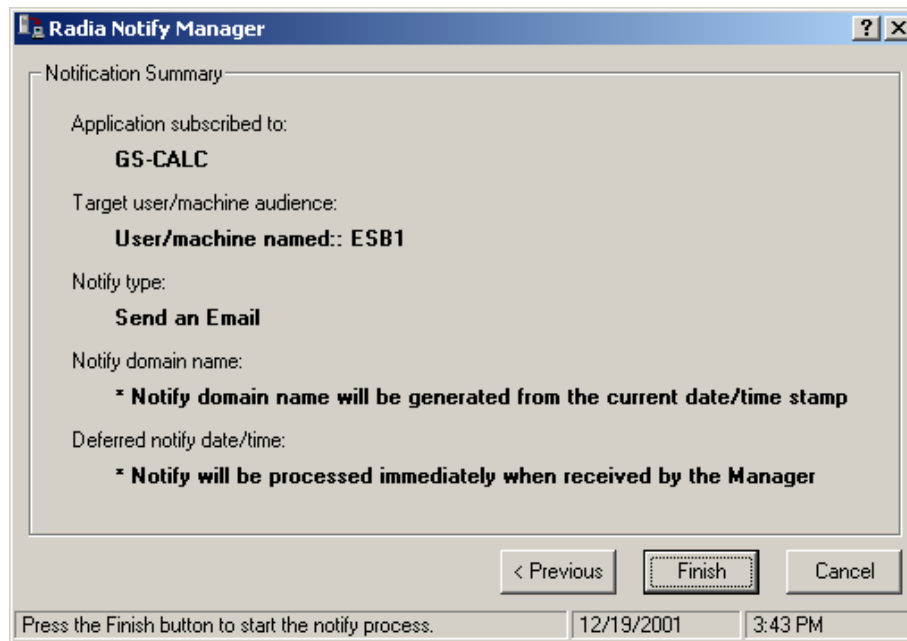
## To defer notification until a specific date and time

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



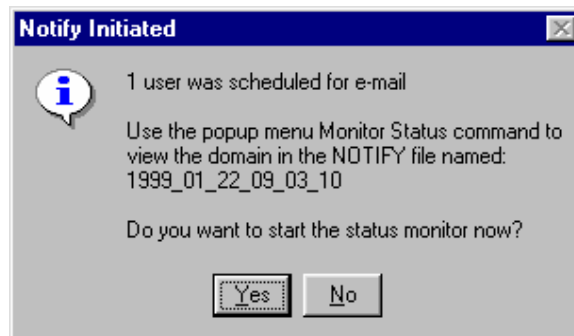
- 2 Enter the date and time when the notify operation should occur. The Configuration Server must be running in order for the notify operation to happen. If not, the notify operation happens the next time the Configuration Server starts, after the designated date and time.
- 3 Click **Next**.

A summary screen opens.



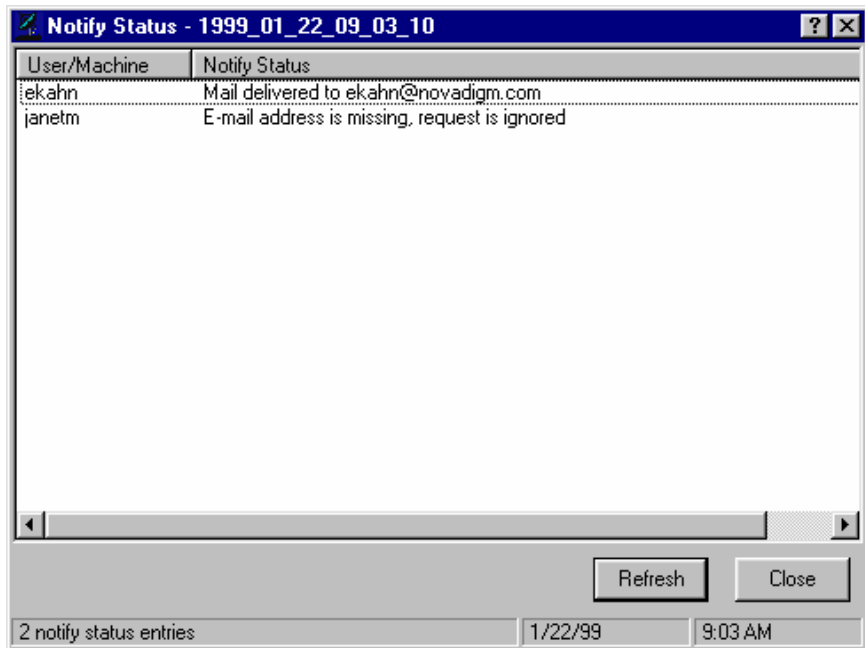
- 4 Click **Finish** to start the Notify.

The 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.



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

A Notify Status window opens.



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

- 6 Select **Refresh** to update the Notify Status window with the latest information. It could take some time for all selected members of the audience list to be notified successfully or unsuccessfully.

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 Configuration Server log and can be viewed there.

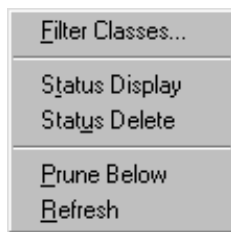
Once a Notify has been initiated for the first time, the NOTIFY file will thereafter appear among the files of the Radia database in the System Explorer tree view.



Use 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.



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

or

Click **Status Delete** to remove from the Radia database the status information that is associated with the Notify operation.

## Notify File Structure

The Notify file is divided into domains, where each domain is one Notify operation. The format of the domain name is YYYY\_MM\_DD\_HH\_MM\_SS, which represents the date and time when the Notify operation was initiated. For example, Figure 55 on page 200 shows an opened NOTIFY domain with the name 1999\_01\_22\_10\_01\_27. This domain holds the results of a Notify operation that was initiated on January 22, 1999 at 10:01:27.

The screenshot shows the 'Radio System Explorer' application. On the left, the 'Database Tree View' shows a hierarchy: Database > NOTIFY > 1999\_01\_22\_09\_03\_10 > 1999\_01\_22\_09\_04\_42 > 1999\_01\_22\_09\_42\_48 > 1999\_01\_22\_10\_01\_27 > NOTIFY > 00000001. On the right, the 'NOTIFY.00000001 Instance Attributes' table is displayed.

Name	Attribute Description	Value
ZUSERID	:<ZUSERID>	ekahn
ZCIPADDR	:<ZCIPADDR>	208.244.225.133
EMAIL	:<EMAIL>	
NTFYTYPE	:<NTFYTYPE>	T
NTFYDATE	:<NTFYDATE>	19990122
NTFYTIME	:<NTFYTIME>	10:01:27
NTFYMSG	:<NTFYMSG>	Successfully notified
NTFYRC	:<NTFYRC>	000
NTFYCMDL	:<NTFYCMDL>	radpinit req="Notify Update",mname=ABC,dname=SOFTWARE,sname=GS-CALC,uid=ekahn
NTFYSUBJ	:<NTFYSUBJ>	
LOCALUID	:<LOCALUID>	
NTFYRTIM	:<NTFYRTIM>	199901522100627099+300
NTFYRNUM	:<NTFYRNUM>	001
NTFYDOMN	:<NTFYDOMN>	1999_01_22_10_01_27
NTFYINS	:<NTFYINS>	00000001
NTFYPORT	:<NTFYPORT>	3465
NTFYPWD	:<NTFYPWD>	EDMPASS
NTFYUINF	:<NTFYUINF>	
NTFYRMAX	:<NTFYRMAX>	007
NTFYDLAY	:<NTFYDLAY>	0300
NTFYCOMN	:<NTFYCOMN>	

**Figure 55: Sample NOTIFY domain entries following an initial notification.**

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 of Figure 55 above), and the results of the Notify operation with respect to that member (see the NTFYMSG and NTFYRC attributes in Figure 55 above ).



## 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 might be turned off, might not be running the Radia Notify module, might not be accessible via the normal communication channel, or might 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 Configuration Server must be started with the Notify Retry Manager (the **ZRTRYMGR** module), as indicated by the last line in the following excerpt from the Configuration Server `edmprof` file.

```
[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexxmgr) 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
```

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 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 simplifies the assigning of notify eligibility to WOL clients. Using the System Explorer, it is easy to connect Notify instances to a **USER**, **WRKGRP**, **DEPT**, and **SERVICE**.

In order to use the drag-and-drop feature to assign clients for WOL Notify, use the System Explorer. Once you have accessed the 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 *Application Manager Guide* for information on how to configure a WOL client for drag-and-drop notify. Refer to the *Configuration Server Guide* for information on changes to the Configuration Server Settings file.

## Filtering—Viewing Portions of the Radia Database

As your distribution model grows, you will find it convenient to restrict the 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 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 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 27: Examples of Valid Filter Expressions**

<b>Filter Specification</b>	<b>Selects</b>
*	Wildcard to select all; removes any filter currently in effect.
_*	Names whose first character is an underscore.

<b>Filter Specification</b>	<b>Selects</b>
*_A*	Names that have a prefix, and any level of the prefix is followed by an <b>A</b> .
*.?LL	Names ending in a period, followed by any one character, followed by <b>LL</b> .
*-??WORD*.HP?	Names containing any number of characters preceding a dash, followed by any two characters, followed by <b>WORD</b> , followed by any number of characters up to a period, followed by <b>HP</b> , followed by any one character.

You can apply filters that remain in effect for an entire System Explorer session (*permanent filters*), or filters that apply only to the current System Explorer window (*temporary filters*).

## Permanent Filters

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

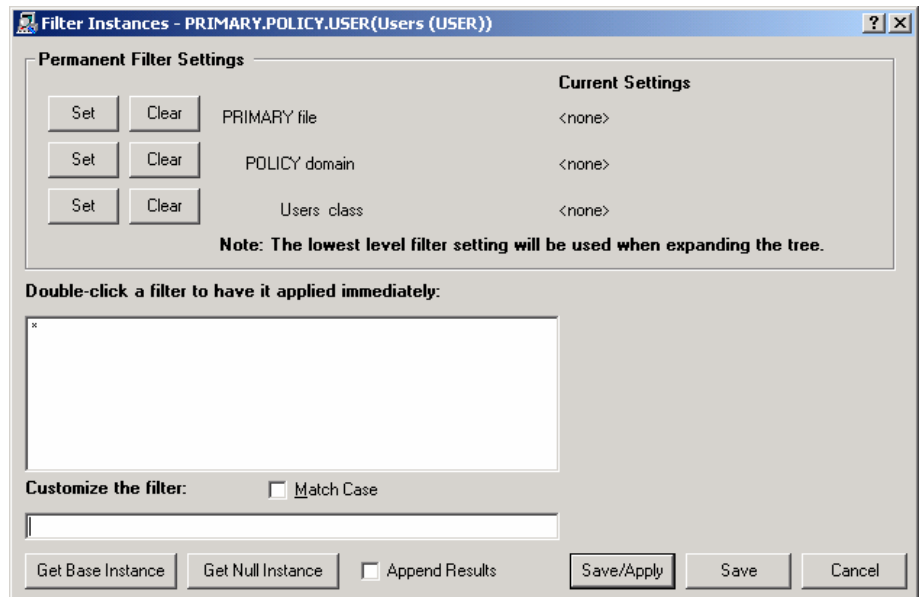
- 2 Scroll down the tree view until you reach the **USER** class.

- 3 Right-click the **USER** class.

The following shortcut menu opens.



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



- 5 Type **ESB\_\*** in the **Customize the filter** text box.
- 6 In the **Permanent Filter Settings** group box, click **Set** next to **Users class**.  
The filter specification is displayed.



### 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 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.

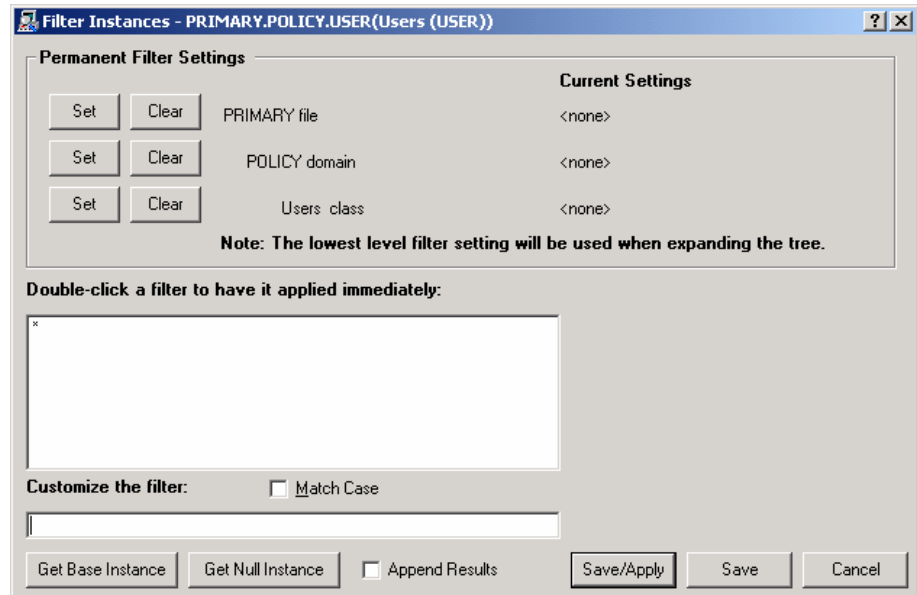
The **Filter Instances** dialog box opens.

- 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 might want to continue viewing the current filter's result, and add an additional filter. To view both results at the same time, enable the **Append Results** checkbox (shown in the following figure) when you perform the next filter.



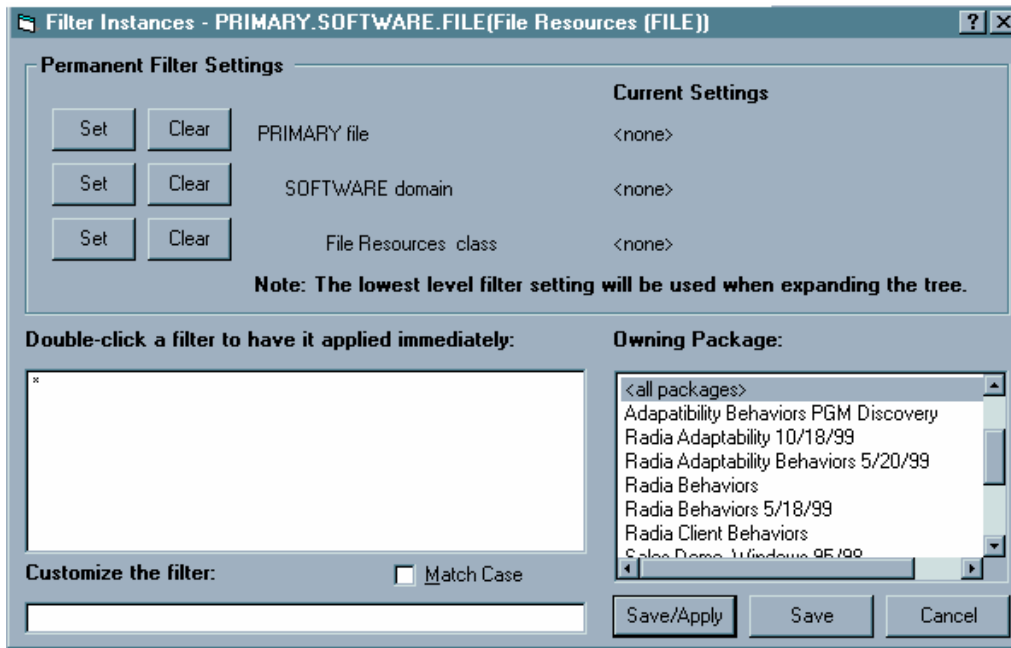
**Figure 56: The Filter Instances dialog box with the Append Results option not selected.**

## Filtering Component Class Instances

Component class instances (such as those of the FILE and DESKTOP classes) can be filtered based on the configuration class instance to which they are connected.

When filtering component classes (such as the FILE class of the SOFTWARE domain in this example), the **Filter Instances** dialog box presents a scrolling list labeled **Owning Package**, as shown in Figure 57 on page 209. The package instances are listed in this scrolling list.





**Figure 57: Filtering component classes, like the FILE class, produces this dialog box.**










To filter FILE instances to select only those 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 System Explorer window.

File Resources (FILE) Class Instances:		
Name	Instance Name	Type
 Amortize Windows 95/98:\Amortize	DABC407A05F8_AD8BF78B	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 58:** 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 anRadia 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 System Explorer window.

## Customizing Express Filters

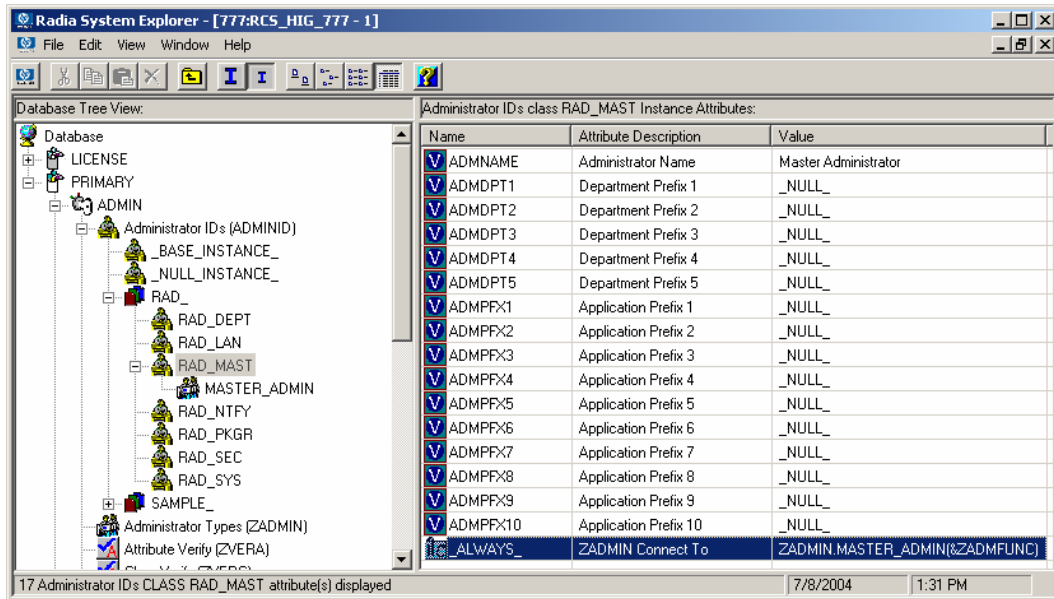
You can specify the lists of express filters that appear for selection in the **Filter Instances** dialog box by storing them in the Radia database, and making any needed connections. The System Explorer retrieves the list of express filters for a particular **Filter Instances** dialog box from the Radia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

The System Explorer locates the correct list in the Radia database by executing a resolution process based on the user ID that was entered when the System Explorer session was started, and the Radia component (i.e., domain, class, or instance) being filtered. See Chapter 6, Understanding the Resolution Process for more information.

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).

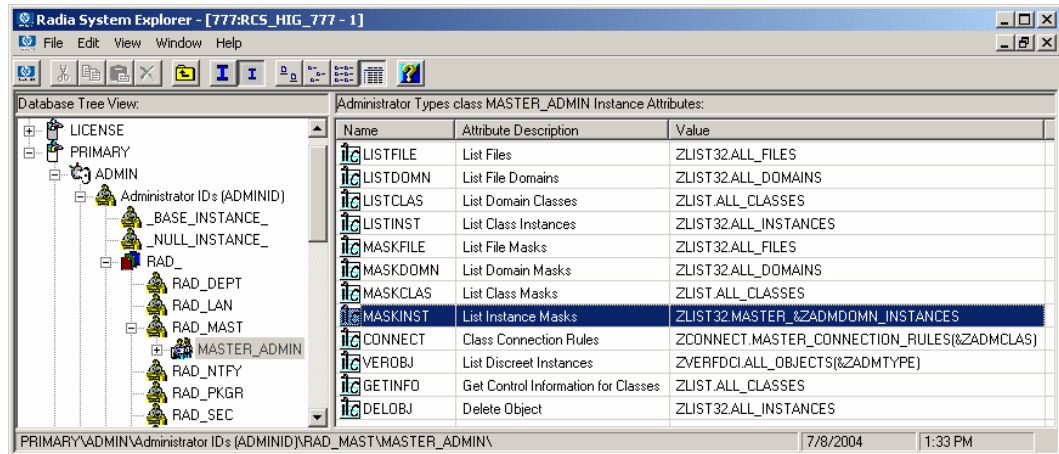
The RAD\_MAST instance contains a connection to an instance of the ZADMIN class, in this case ZADMIN.MASTER\_ADMIN.



**Figure 59: 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.**

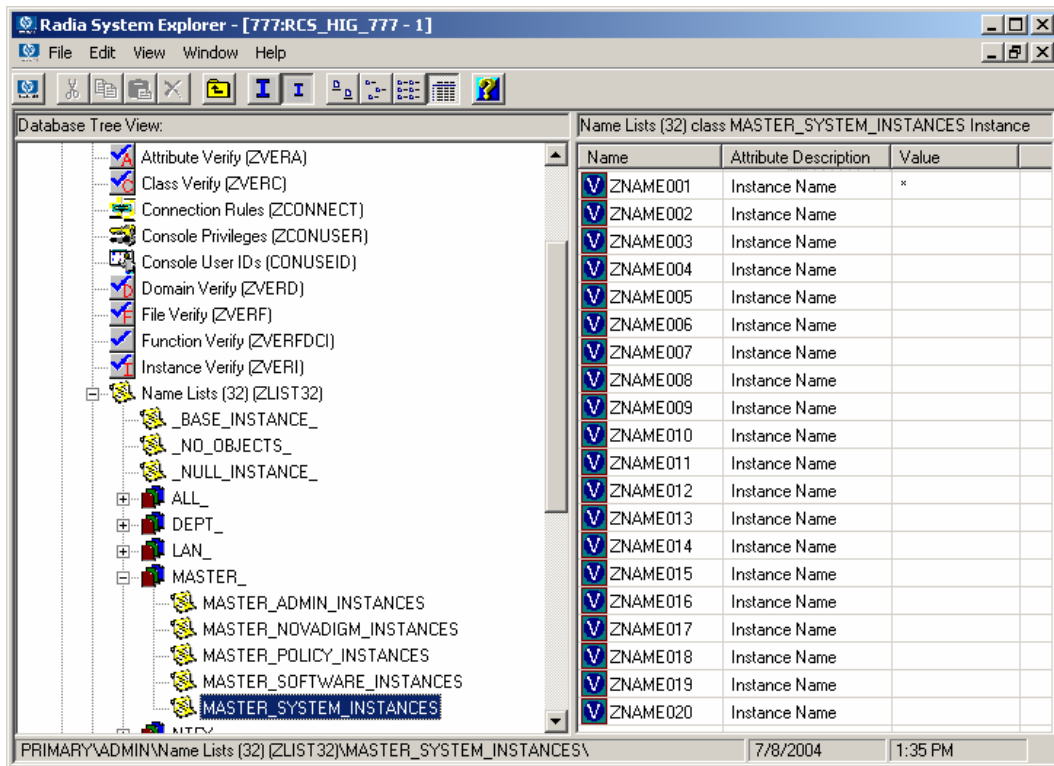
The connection to ZADMIN.MASTER\_ADMIN passes a parameter in the system message (&ZADMFUNC) that indicates which of the connections contained in the ZADMIN.MASTER\_ADMIN instance should be made. Only those connections whose names match the value of the system message (or whose name is \_ALWAYS\_) are actually made during a particular resolution execution. The System Explorer sets the parameter's value before initiating the resolution. In this case, the 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 60: 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 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 61: A connection is further made to LIST32.MASTER\_SYSTEM\_INSTANCES.**

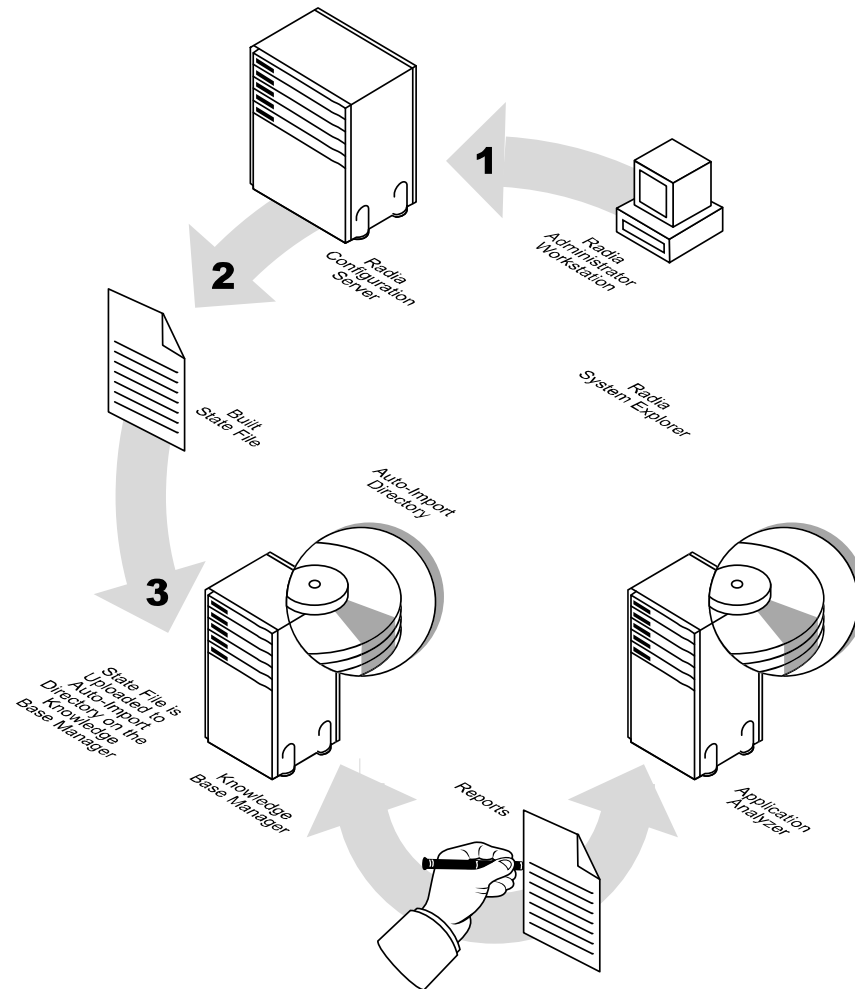
To change the list of express filters for a particular domain, edit the appropriate instance of the ZLIST32 class. The changes will affect all administrators whose user IDs connect to the ZLIST32 instance you modify.

To set custom express filters for particular administrators, use the information detailed above to connect their user ID instances in the ADMIN.ADMINID class to the appropriate ZLIST32 instance containing the desired list of express filters.

See Adding an Instance on page 88 and Editing an Instance on page 90 for information on how to use the System Explorer to make these changes.

# Building State Files with the System Explorer

Use the System Explorer, **Build State File** function to export an existing Radia Service as a set of state files into the Radia Knowledge Base.

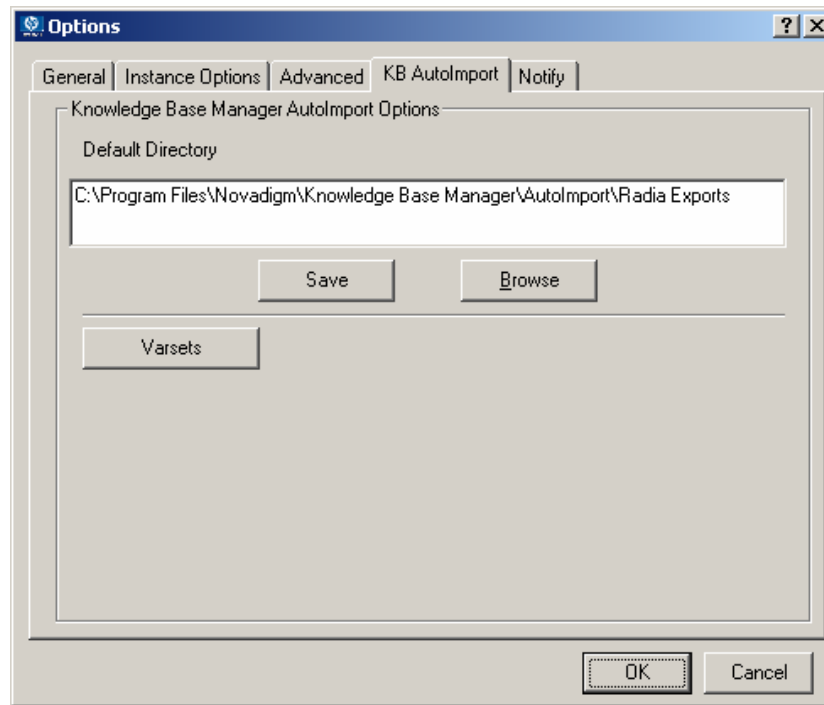


**Figure 62: State file export process.**

Multiple state files can be built simultaneously. A separate state file is created in the Radia Knowledge Base for each combination of operating system and package connection that the application service supports.

▶ The **Build State File** function is designed for native packages for Windows operating systems. If you attempt to export a non-Windows service, you will receive an error message.

Use the **KB AutoImport** tab on the System Explorer **Options** menu to define the Knowledge Base Manager default auto-import directory.



**Figure 63: KB AutoImport tab of View Options menu.**

Click **Save** on the **KB AutoImport** tab to ensure that the default location of the Knowledge Base Manager default auto-import directory is set.

## State File Creation Process

The state file creation process is accomplished in two phases; *discovery* and *execution*. During the discovery phase, temporary values are generated for any variables that are found within the package that are not defined in the operating system specific `.ini` file (this `.ini` file is located in the `varsets` subdirectory of the state file folder). During the build process, you are prompted to enter a more meaningful value for these variables.

As stated above, all variable values are defined within the associated operating system `.ini` file located within the `varsets` directory. When a value cannot be found during the discovery phase, a temporary variable is created, but in certain circumstances, the temporary variable might not be created in the context intended and the build process will fail. This occurs only when a specific value has been assigned to a variable present within a ZSTOP expression within your package, for example, resolving an expression based on a specific memory size. To ensure the build process completes successfully, add any special case variable values to the `.ini` file before creating the state file.

### Example

An application contains a package level ZSTOP expression such as:

```
SPACE (TRANSLATE (WORD (EDMGETV (ZCONFIG, ZHDWMEM), 1), , ", " ), 0) <
131072000
```

In this case, the variable `ZCONFIG.ZHDWMEM` must be 128MB (131072000) or more for the package to deploy. By adding the item `ZHDWMEM` to the `ZCONFIG` section of the operating system-specific `.ini` file, the discovery process will continue correctly. The updated `.ini` file will contain the following:

```
[ZCONFIG]
ZHDWMEM = 131072001
```



In the example above, the value added to the `.ini` file is one (1) larger so that the expression (as defined above) will fail allowing resolution to continue for this package.



## To build a state file using the System Explorer

- 1 Start the Radia System Explorer.
- 2 Double-click **Primary**.
- 3 Double-click **Software**.
- 4 Double-click **Application**.
- 5 Right-click the Application or select and right-click the Applications for which you would like to build state files.

▶ Select multiple applications by holding the **Ctrl** key on the keyboard and clicking the names of the applications.

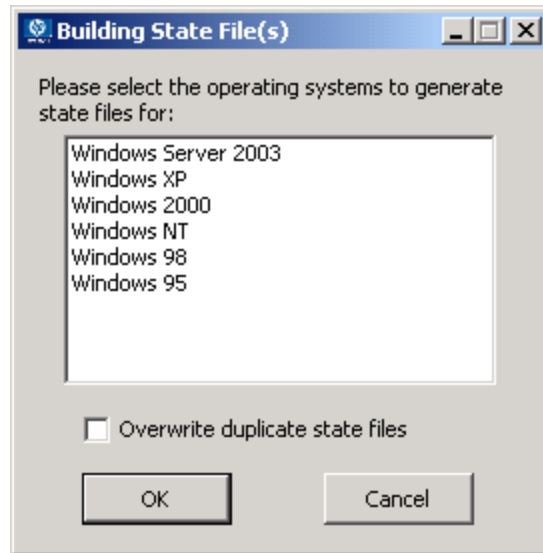
Use the **Shift** key to select a group of applications.



**Figure 64: Application context menu – Build State File option.**

- 6 Select **Build State File** from the shortcut menu.

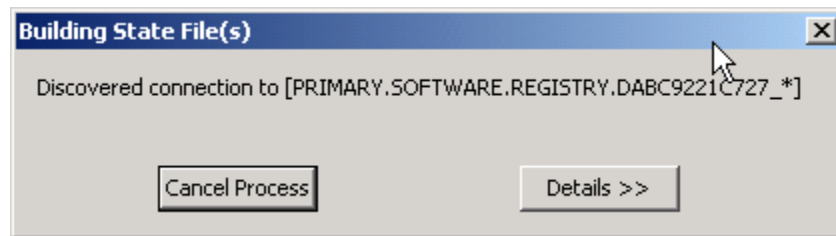
The **Build State File(s)** window opens.



**Figure 65: The Build State File(s) window.**

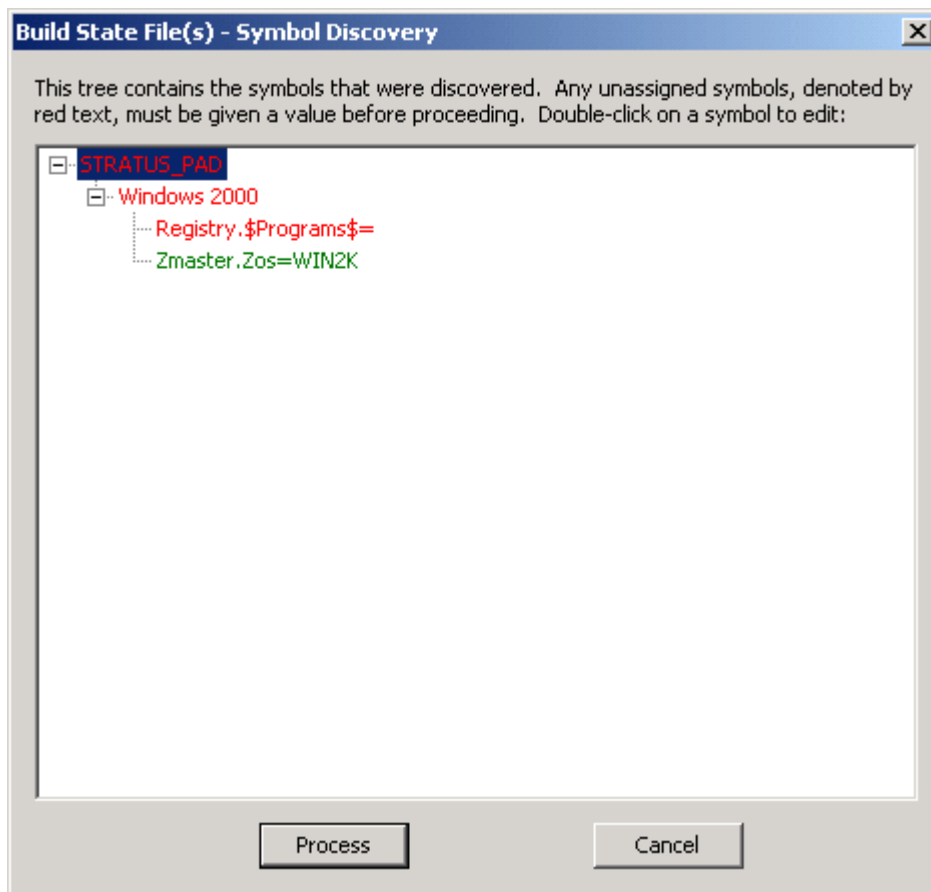
- 7 Select the operating system for which you would like to generate the state file.
  - To overwrite existing state files with an updated version, select the **Overwrite duplicate state files** check box.
- 8 Click **OK**.

The state file build process begins.



**Figure 66: The Building State File(s) message.**

When the process is finished, a window will open displaying any symbols that were discovered.



**Figure 67: Symbol Discovery window.**

- 9 Expand the tree view to display all the symbols that were discovered. Double-click each symbol displayed in red to assign an appropriate value before proceeding. Any symbols that you define here are stored in an OS-specific `.ini` file in the `varsets` folder. For example, if any values were defined for a Windows 2000 state file, they would be stored in the `varsets` directory in a file called `win2k.ini`.
- 10 Click **Process** when finished.

A processing window will open.

- 11 When the processing is complete click **OK**.

During the processing, a state file is built for the selected packages in the Knowledge Base Manager automated-import directory. If this directory is being monitored by the Knowledge Base Manager service, then the state files and associated Radia Service entries are imported into the Radia Knowledge Base.

Use the HP OpenView Configuration Analyzer using Radia (Configuration Analyzer) to view the exported state-file data. This is discussed in the *Installation and Configuration Guide for the HP OpenView Configuration Using Radia (Configuration Analyzer Guide)*.

## 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 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 System Explorer offers the ability of setting filters, and appending filter results, to provide user-defined restricted views of the Radia database.
- The Build State File option enables you to export any application file from the System Explorer, build a state file, and have it automatically imported into the Radia Knowledge Base.



---

## 6 Understanding the Resolution Process

At the end of this chapter, you will:

- Understand the resolution process of the HP OpenView Configuration Server Using Radia (Configuration Server).

## The Resolution Process

The 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 HP OpenView Using Radia (Radia) database and parameters included in the request. In other words, what Radia does depends on the information that is stored in its database, and the information that accompanies the request for Radia to perform an action.

For example, the Radia client-connect submits service requests to the Configuration Server, and the Configuration Server performs a resolution process in response to each such request.

## Heaps and Objects

One way that the 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 (such as ZMASTER) can consist of a single instance or several instances, as does the FILE object.

You can inspect and even modify desktop objects using the HP OpenView Client Explorer Using Radia (Client Explorer).

## Client Explorer

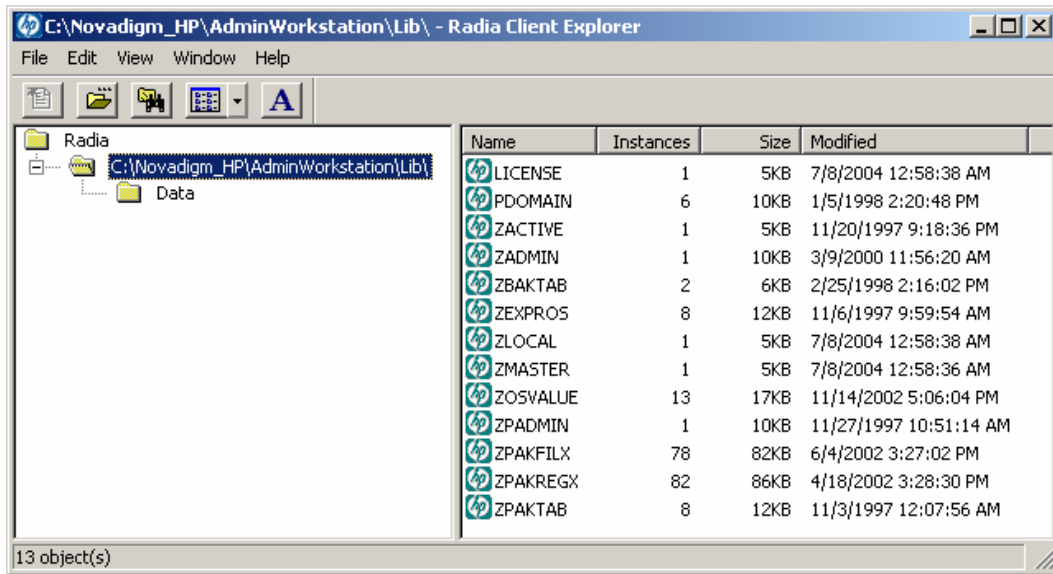
The Client Explorer is used to view or edit objects. A user with the Administrator Workstation installed can use the Client Explorer as a diagnostic utility to view local objects on the desktop, edit objects, or create new objects.

### To access the Client Explorer

- Click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia Client Explorer**.

The **Radia Client Explorer** opens.



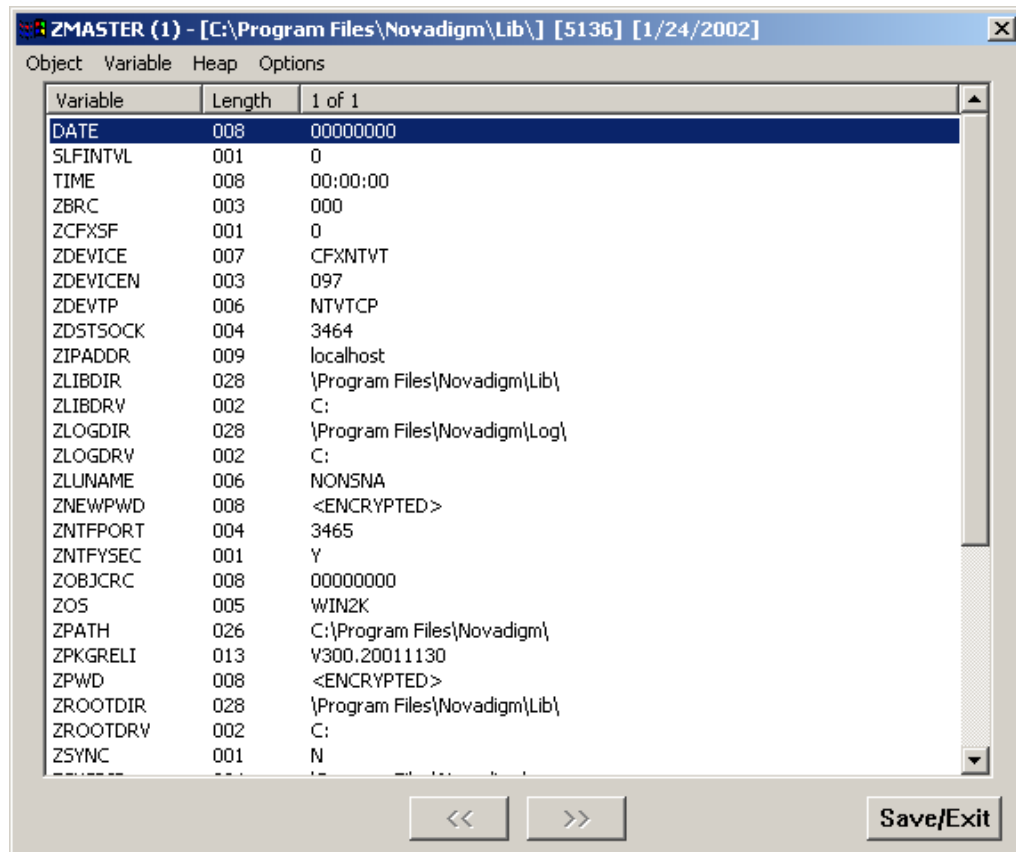


**Figure 68: The list view in the Client Explorer.**

The figure above displays the Client Explorer at its highest level. The list view (the right side of the window) in the Client Explorer displays the:

- Instance name,
- Number of heaps (Instances),
- Size of the instance, and
- Date and time of the last modification.

To view an instance, double-click it. The instance information will be displayed as shown in the following figure.



**Figure 69: 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 each 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 System Explorer.

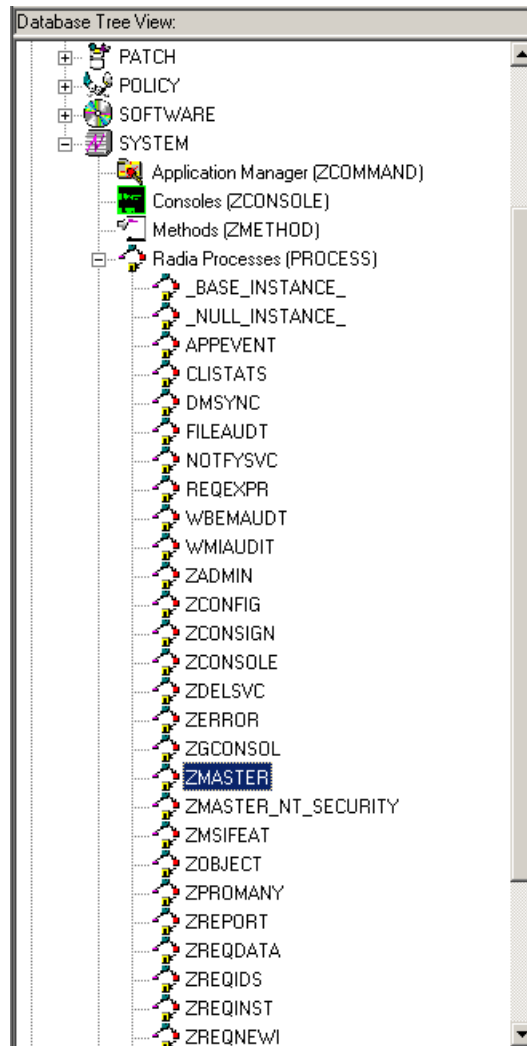
## Resolution Process—Where Does It Begin?

As previously stated, the resolution process begins when a client computer connects to the Configuration Server. The ZMASTER is the first object sent

to the Configuration Server during the client connect. It contains information about the client computer that is needed to run Radia, such as the identity of the subscriber and the IP address of the computer.

The 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 Configuration Server finds the PROCESS instance for the ZMASTER. This is known as the **process entry point**. Its location is SYSTEM.PROCESS.ZMASTER.



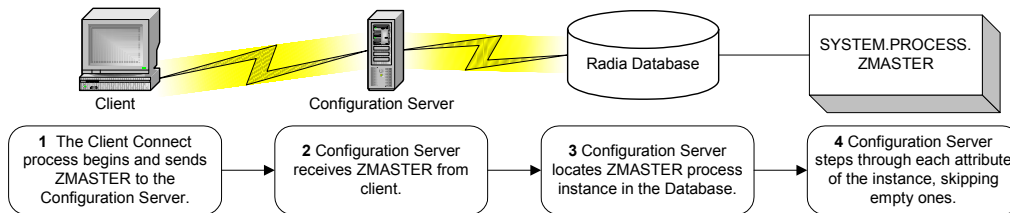
**Figure 70: SYSTEM.PROCESS.ZMASTER, the location of the ZMASTER object in the Radia database.**

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

- Set variable values.

- Evaluate an expression.
- Execute a method.
- Connect to other instances.

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



**Figure 71: The Configuration Server performing resolution.**

## Resolution Example

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

In Figure 72 on page 230 the first connection instance links to `POLICY.USER.&(ZMASTER.ZUSERID)`.

After processing this, 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 &ZCURI OBJ
ZMAXOKRC	Max acceptable method Return Code	008

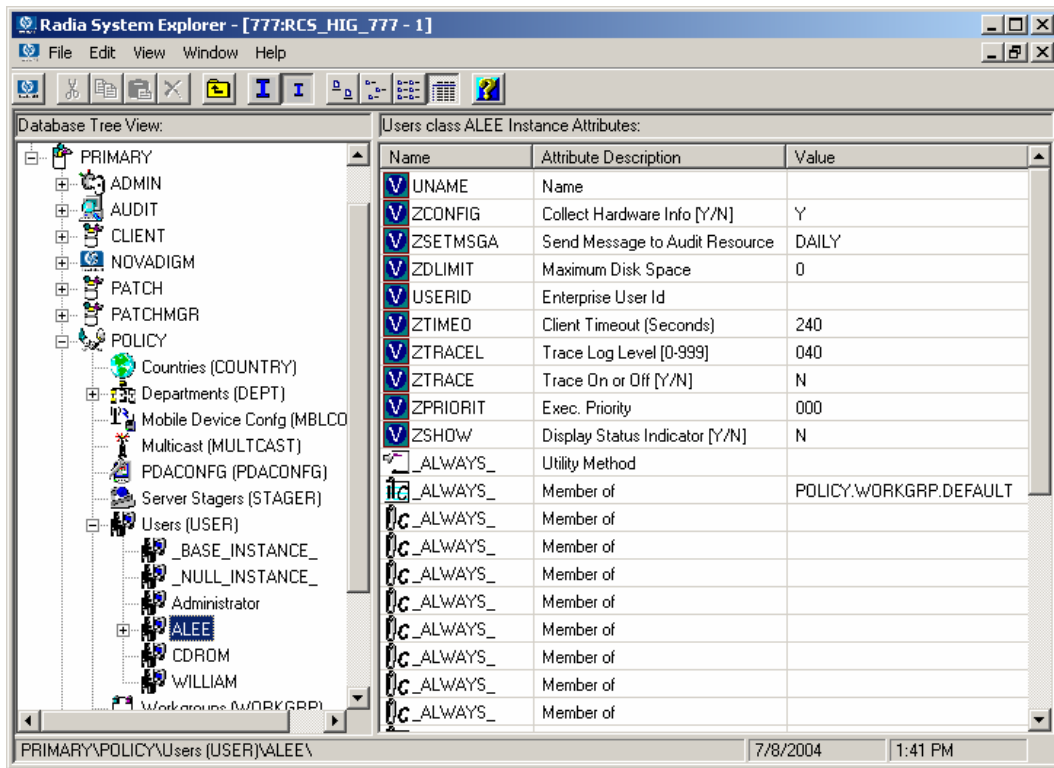
**Figure 72: The PRIMARY.SYSTEM.PROCESS.ZMASTER instance.**

## Resolution Process and Symbolic Substitution

During resolution, the 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.

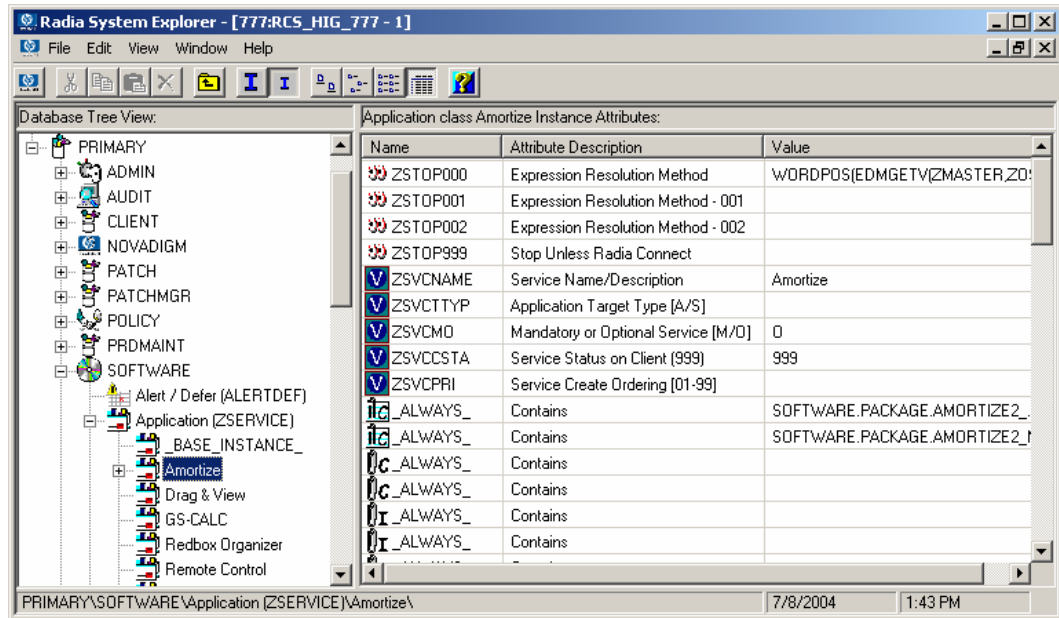
References to be processed with symbolic substitution are specified using an initial ampersand.

For example, in SYSTEM.PROCESS.ZMASTER there is an always connection to POLICY.USER.&(ZMASTER.ZUSERID). The Configuration Server substitutes the value of ZUSERID from the ZMASTER object that is in global memory. In Figure 73, the value of ZUSERID is ALEE. The resolution process will connect to POLICY.USER.ALEE to begin resolving the ALEE instance.



**Figure 73: POLICY.USER.ALEE instance.**

In Figure 74, after setting a number of variables, the first connection attribute is to SOFTWARE.ZSERVICE.AMORTIZE. In your implementation, the POLICY instance might connect to a workgroup or a department that would then connect to a service.



**Figure 74: The ZSERVICE.AMORTIZE instance.**

The resolution proceeds to the service instance that is indicated in the first connection of the USER as shown in Figure 74. The Amortize service resolution begins with a ZSTOP expression variable. This expression variable contains statements that, if evaluate to *true*, stop the resolution of the current instance. An expression allows alternative paths to be taken in a given resolution, based on 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 a different 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 72 on page 230, the next attribute connects to the PUTPROF\_ZMASTER method.



**A method** is a program that performs functions based on certain parameters.





## Summary

- The 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 Configuration Server recognizes a service request is when it receives an object.
- The 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 Configuration Server during the Client Connect.
- During resolution, the Configuration Server performs symbolic substitution to set values and connect to other instances.

---

## 7 Maintaining Connection Rules in the Radia Database

At the end of this chapter, you will:

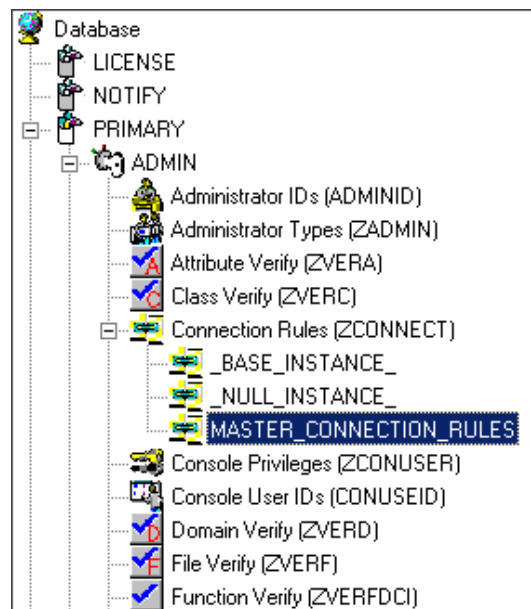
- Understand connection rules.
- Know how to enable connection rules.

# Introduction

The 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

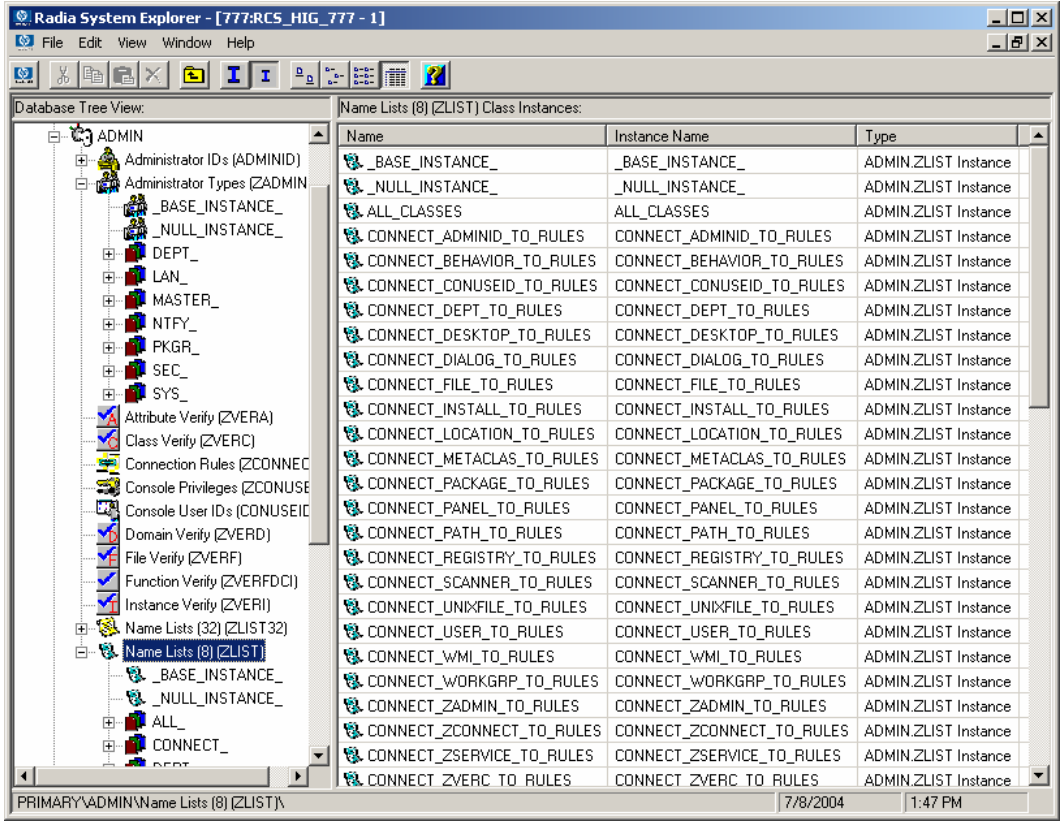
In order to allow for drag-and-drop capabilities for an added class, it is necessary to establish rules for the new class to follow. These connection rules are stored in the PRIMARY.ADMIN domain in the **MASTER\_CONNECTION\_RULES** instance of the **Connection Rules (ZCONNECT)** class.



**Figure 76: 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

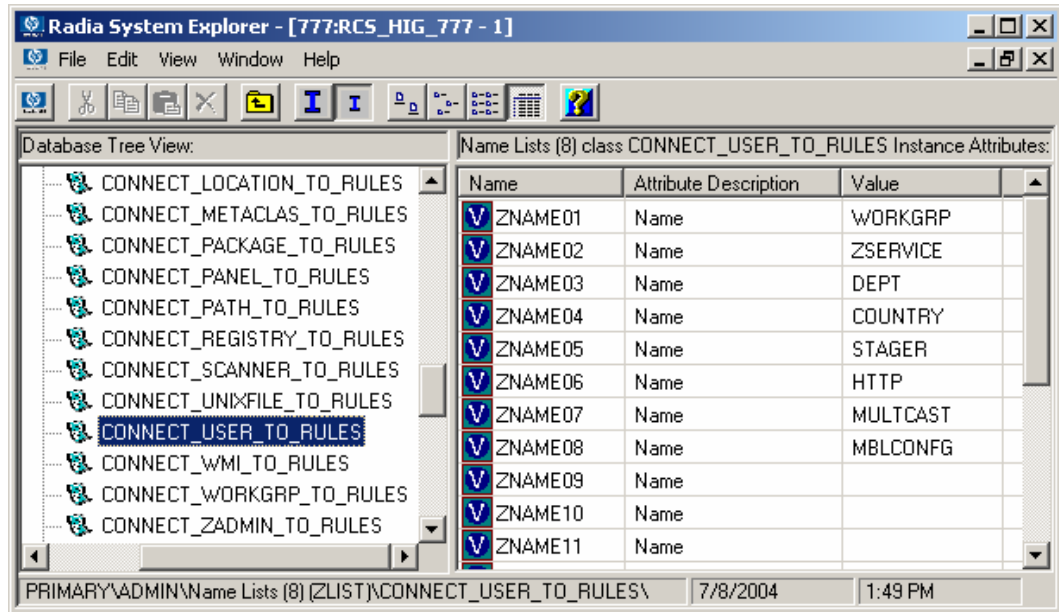
a **connection rule**, you will have to first create the rule in the ZLIST class. Then you will add the rule to the ZCONNECT class.



**Figure 77: The Name Lists (ZLIST) class in the ADMIN domain.**

The 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 78.



**Figure 78: The ZLIST.CONNECT\_USER\_TO\_RULES instance.**

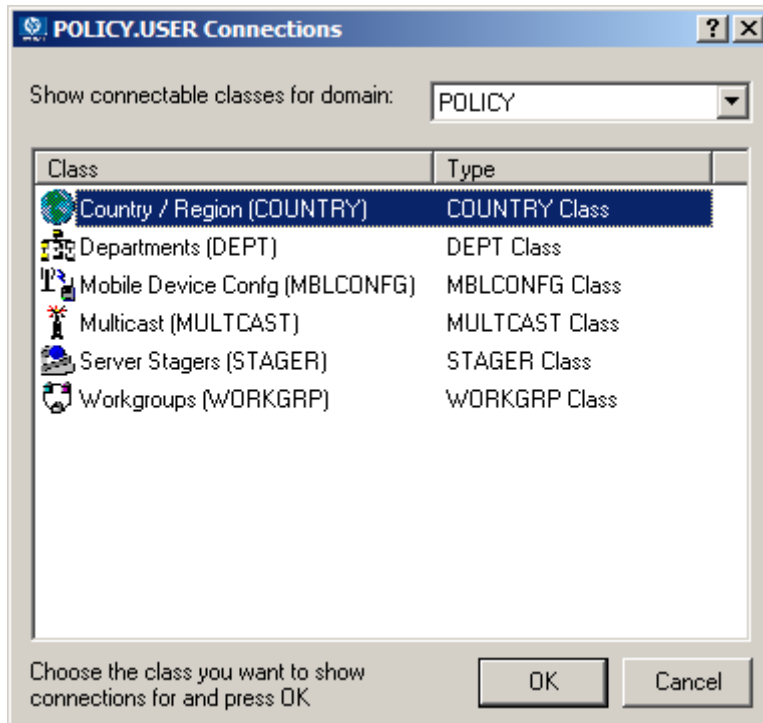
An instance of this particular USER class can contain a connection to any of the following classes:

- WORKGRP
- ZSERVICE
- DEPT
- COUNTRY
- STAGER
- HTTP
- MULTICAST
- MBLCONFG

The ZLIST instance contains a list of up to 20 classes to which the class, represented by the ZLIST instance (in this case, USER), can connect. You can edit this list to include additional classes, or remove classes from a connection rule. Of course, if any class in the 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 79 shows the connectable classes for the POLICY domain where the USER class is located.



**Figure 79: The connectable classes for the POLICY domain.**

## Enabling Connections in New Classes

If you add to the Radia database a new class 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.

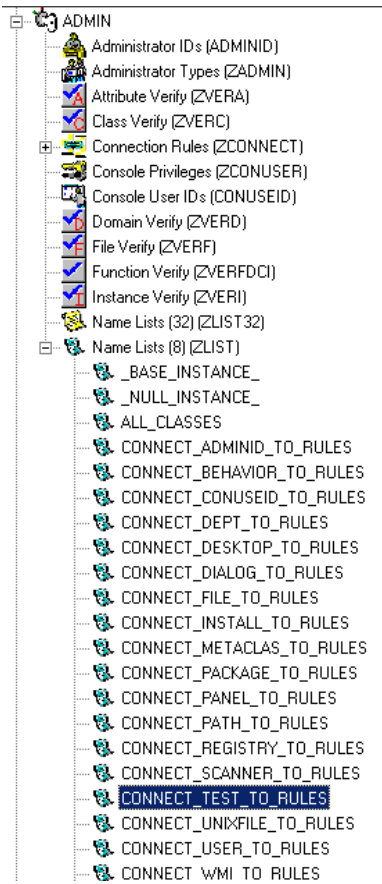
- Set the new class's variable in the MASTER\_CONNECTION\_RULES.

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 new class, for example, TEST.
- 1 Open the System Explorer and navigate to the **PRIMARY.ADMIN.Name Lists (ZLIST)** class.
  - 2 Right-click **ZLIST** and select **New Instance**.
  - 3 Name the new instance as suggested above, `CONNECT_TEST_TO_RULES`.

Figure 80 on page 241 shows the added `CONNECT_TEST_TO_RULES` in the ZLIST class.

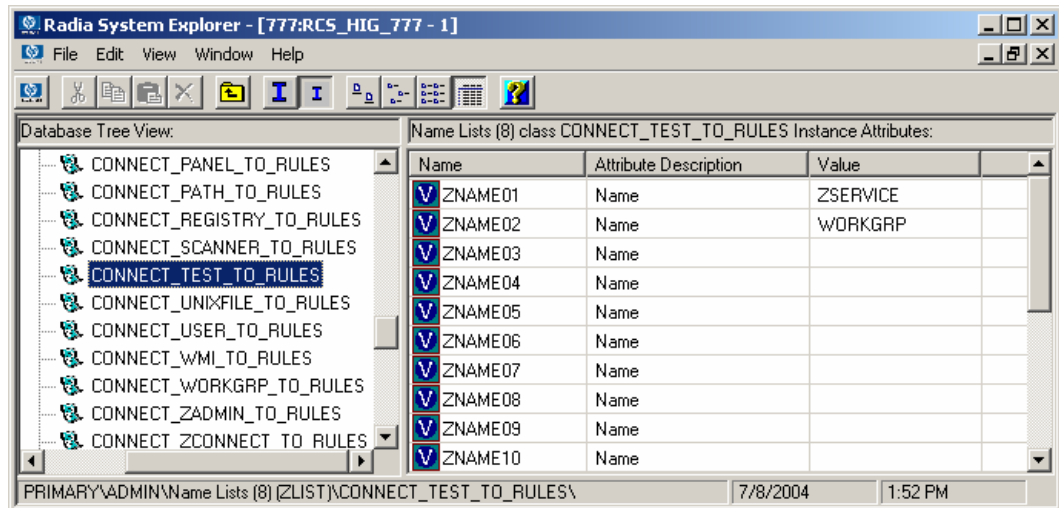




**Figure 80: CONNECT\_TEST\_TO\_RULES in the ZLIST class.**

- 4 Edit the CONNECT\_TEST\_TO\_RULES instance, setting each ZNAME $nn$  variable to the name of a class to which the new class can be connected.

For example, set ZNAME01 to allow connections to the ZSERVICE and WORKGRP classes, as shown in Figure 81.



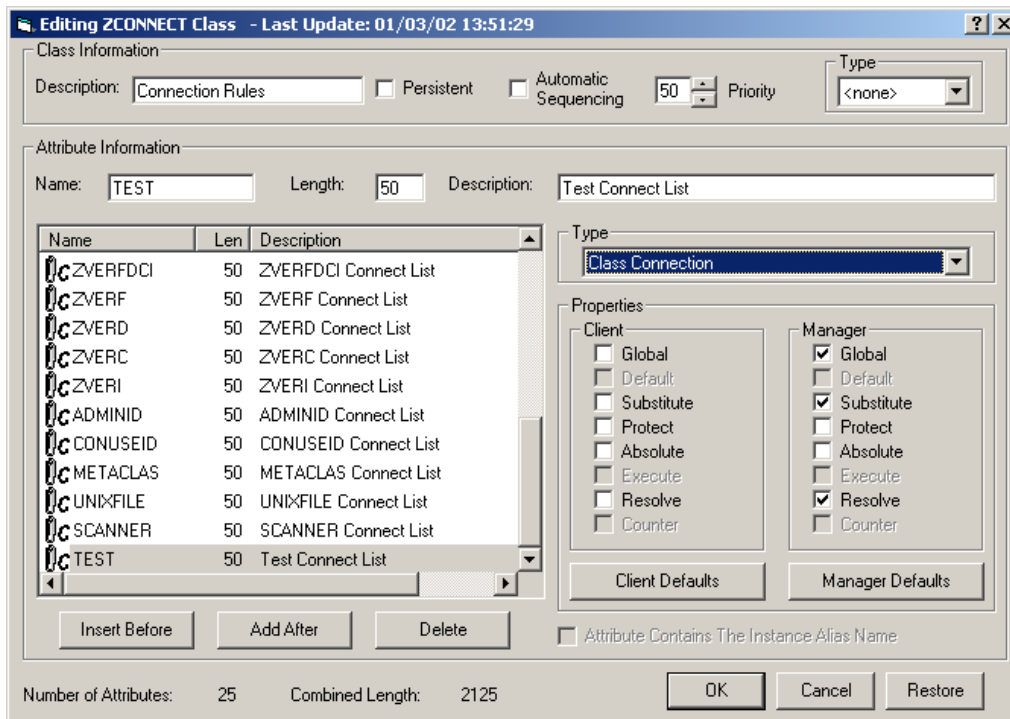
**Figure 81: Connectable classes for the TEST class.**

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

- 5 Navigate to the **PRIMARY.ADMIN.ZCONNECT** class.
- 6 Right-click on the **ZCONNECT** class and select **Edit Class** from the context menu.
- 7 Add the new class variable (TEST) to the class template for the ZCONNECT class. The length of the attribute will be **50**, and the description is **Test Connect List**.

➤ When adding an attribute to the ZCONNECT Class, be sure to change the attribute type to **Class Connection**. See Figure 82 on page 243.

➤ For information on editing a class, see **Editing a Class** beginning on page 72.

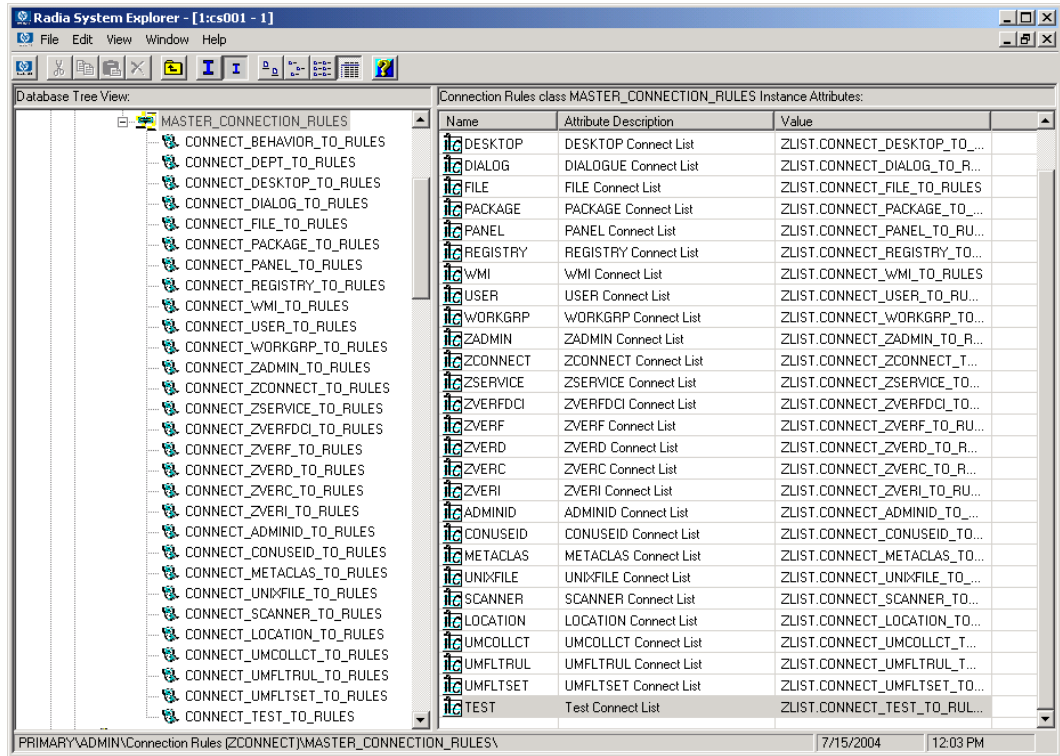


**Figure 82: Add the new class's name to the list of variables in the Connection Rules class.**

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.

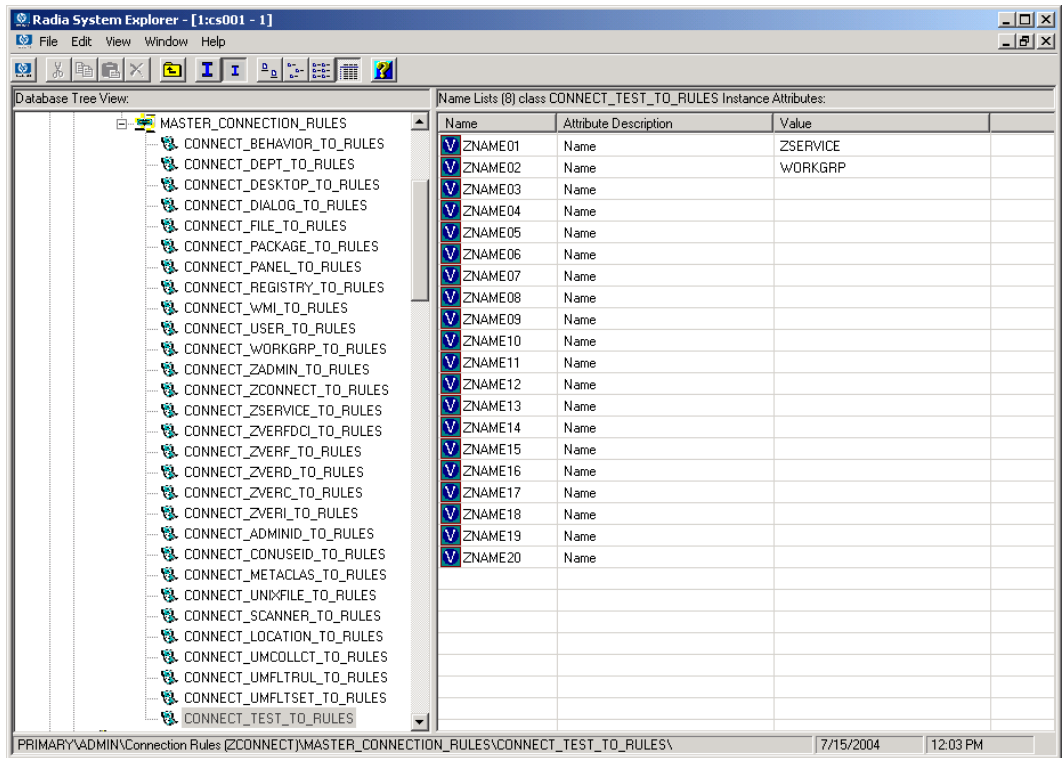
- 8 Locate the new instance in the list view (right side of the screen), and set the value of the newly added variable to ZLIST.CONNECT\_TEST\_TO\_RULES where *newclass* is the name of the newly added class

For example, check the value that has been set for the TEST attribute in Figure 83 on page 244.



**Figure 83: ZLIST.CONNECT\_TEST\_TO\_RULES set as the value of ZCONNECT.MASTER\_CONNECTION\_RULES.TEST.**

Figure 84 on page 245 shows that the new connection rule has been added to the MASTER\_CONNECTION\_RULES, thereby allowing drag-and-drop connections for the new class.



**Figure 84: The completed connection rule in MASTER\_CONNECTION\_RULES for the TEST class.**

## Summary

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

---

## 8 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 to manage feature sets in native MSI mode.

## Feature Set Editor Overview

The System Explorer includes a Feature Set Editor, with a wizard type interface, to guide 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 (Microsoft Windows Installer) for installing and maintaining software products on the Windows platform. Radia products can install and manage Windows Installer-enabled applications as follows:

The HP OpenView Publisher Using Radia (Publisher) allows for *basic* and *advanced* methods of publishing and configuring Windows Installer-enabled Applications.



For information regarding the packaging of Windows Installer-enabled applications, see the *Getting Started Guide for the HP OpenView Extensions for Windows Installer Using Radia (Extensions for Windows Installer Getting Started Guide)*.

For information regarding the publishing of Windows Installer-enabled applications, see the *Publisher Guide for the HP OpenView Administrator Workstation Using 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.

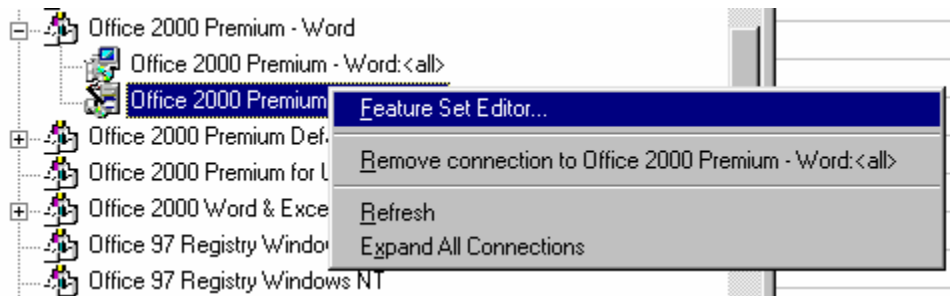
Radia provides the ability to discover and manage previously installed Windows Installer-enabled applications that were not deployed using Radia.



# Accessing the Feature Set Editor

To access the Feature Set Editor

- 1 Locate the Application Packages instance that represents the software package in the Radia database whose features you wish to edit.
- 2 Double-click this instance to display its MSI and MSIFEATS connections.
- 3 Right-click the MSIFEATS connection in the tree view, as follows:



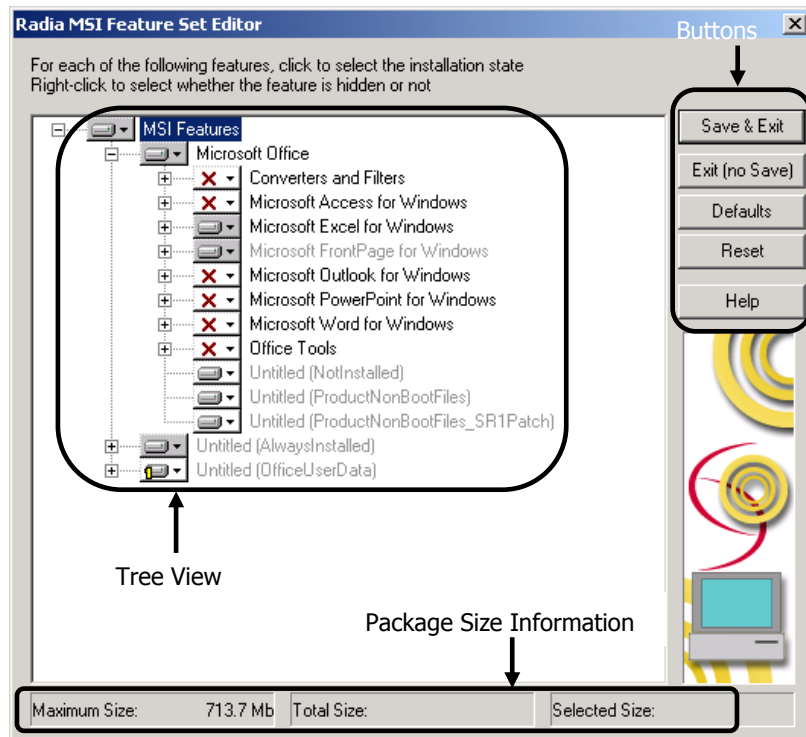
**Figure 85: Accessing the Feature Set Editor thru the System Explorer tree view.**

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

The Feature Set Editor dialog opens.

## About the Radia MSI Feature Set Editor

Figure 86 shows that 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.

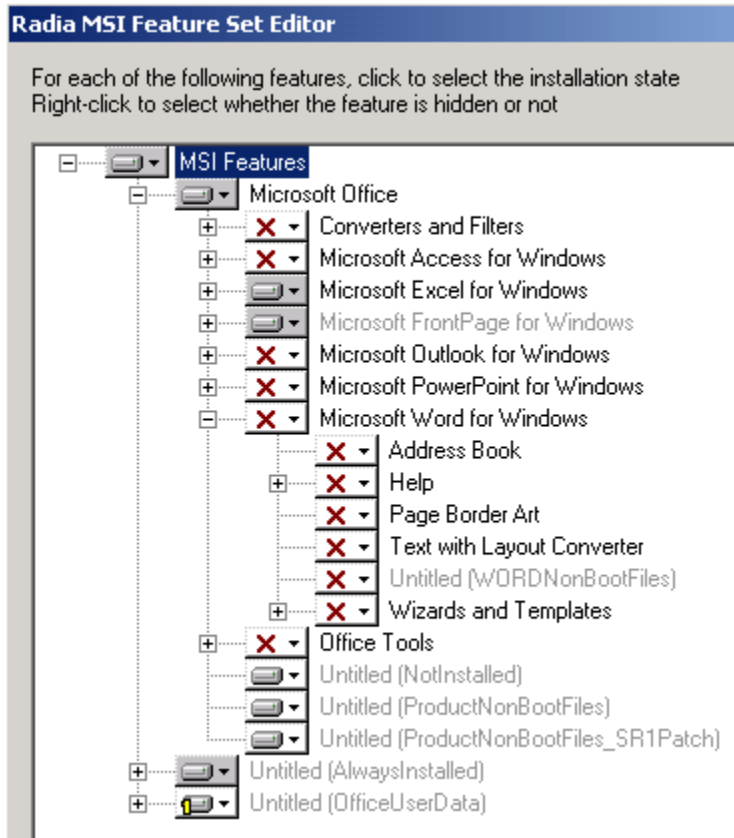


**Figure 86: The Feature Set Editor dialog box.**

## Tree View

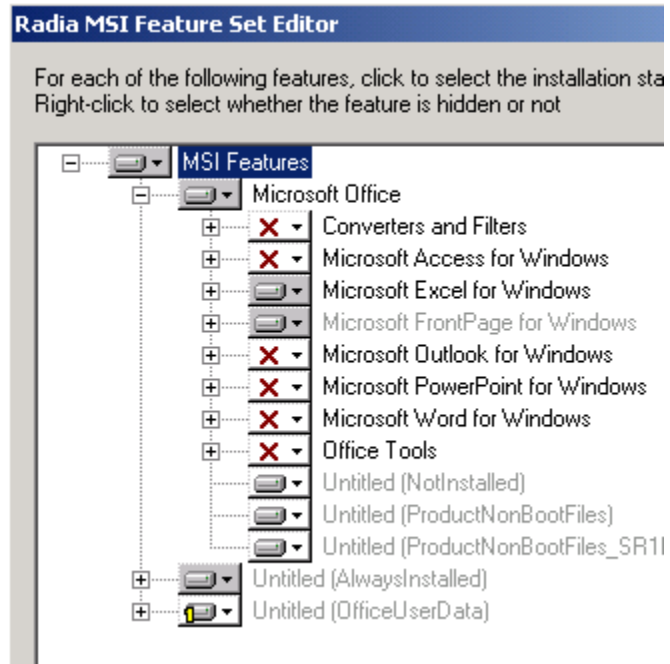
Features are grouped into a hierarchy that you navigate using the tree control. Higher-level features consist of two or more lower-level (i.e., child) features.

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 87: Child features displayed.**

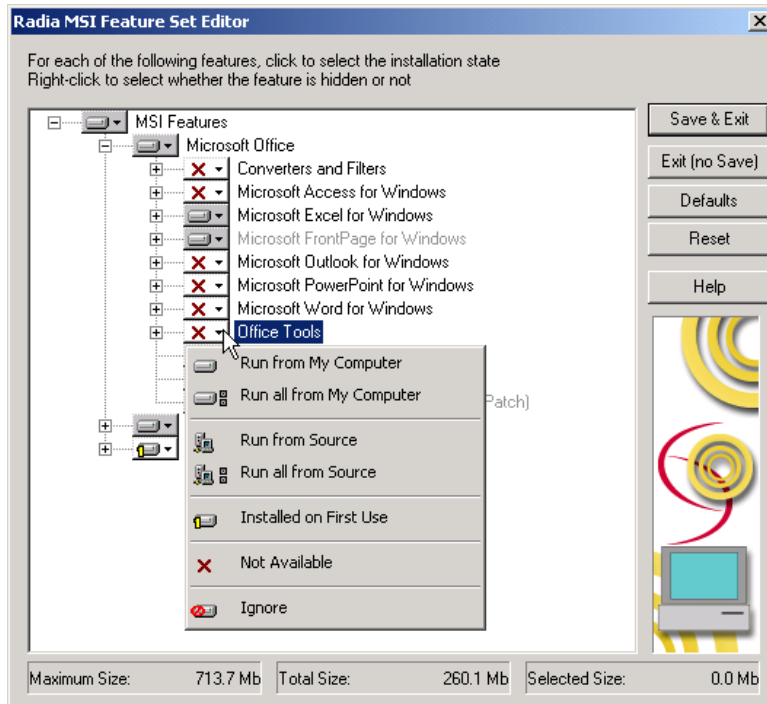
To close the display of child features beneath a particular feature, click on the minus sign in the small box next to the selected feature:



**Figure 88: Child features closed under Microsoft Word.**

## Features Menu

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







**Figure 89: 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 28.

**Table 28: Features Menu Options**

Feature Icon and Name	Description of Function
 Run from My Computer	Features with this attribute are physically installed on a hard drive on the client computer, and therefore perform best.

Feature Icon and Name	Description of Function
	<p>When you select this attribute for a feature, the attribute is automatically applied to all features that are children of the selected feature, unless the child feature's attribute is already set to <b>Installed on First Use</b> or <b>Not Available</b>.</p> <p>The selected feature's children appear on branches of the tree, and are beneath and connected to the selected feature.</p>
 Run all from My Computer	<p>Features with this 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 <b>Run from My Computer</b>, and they, too, are physically installed on the client computer.</p> <p>Features with this attribute perform best 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 <b>Run from My Computer</b> or <b>Run all from My Computer</b>. 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 might not be available at that time.</p>
 Run from Source	<p>Features with this 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 attribute is automatically applied to all features that are children of the selected feature, unless the child feature's attribute is already set to <b>Not Available</b>.</p> <p>The <b>Run from Source</b> and <b>Run all from Source</b> attributes save local hard drive space at the expense of slower performance and the requirement of having an active connection to the AIP.</p>
 Run all from Source	<p>Features with this attribute are not physically installed on a hard disk on the client computer, rather they are run from the AIP. This is normally a network share persistently accessible to the client computer.</p> <p>When you select this attribute for a feature, the <b>Run from Source</b> 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>

Feature Icon and Name	Description of Function
	<p>The <b>Run from Source</b> and <b>Run all from Source</b> attributes save local hard drive space at the expense of slower performance and the requirement of having an active connection to the AIP.</p>
 Installed on First Use	<p>Features with the <b>Installed on First Use</b> 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 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 they first invoke it, which might surprise and displease them.</p> <p>When you select this attribute for a feature, the <b>Installed on First Use</b> attribute is applied automatically to all features that are children of the selected feature, unless the child feature's attribute is already set to <b>Not Available</b>.</p> <ul style="list-style-type: none"> <li>• See the Important Note on page 257 for additional information.</li> </ul>
 Not Available	<p>Features set to <b>Not Available</b> are not installed, are unavailable to the end user and could affect other services. The attributes for all child features are also set to <b>Not Available</b>.</p> <p>For example, one service can be defined that deploys Microsoft Word, and all other features of Office 2000 are set to <b>Not Available</b>. Another service can be defined that deploys Microsoft Excel, and all other features of Office 2000 are set to <b>Not Available</b>.</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 <b>Not Available</b>. Therefore, Microsoft Excel is installed and</p>

Feature Icon and Name	Description of Function
	<p>Microsoft Word is uninstalled.</p> <p>If the end user selects an option that requires the feature, the software instructs the end user to run the Setup again and change the installation state of the feature. However, end users whose software is HP-managed generally don't have access to the Setup.</p> <p>Assign the <b>Not Available</b> 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><b>Note:</b> 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 <b>Ignore</b> 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 <b>Ignore</b>.</p> <p>For example, one service can be defined that deploys only Microsoft Word, with all other features of Office 2000 set to <b>Ignore</b>. Another service can be defined deploying Microsoft Excel and PowerPoint, setting all other features of Office 2000 to <b>Ignore</b>. 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>





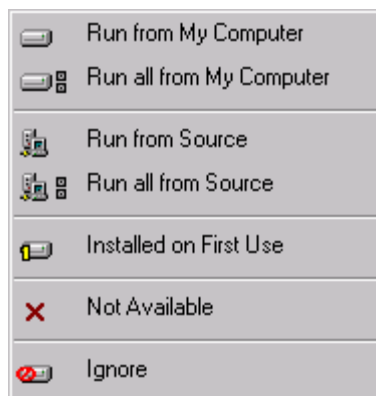
### Important Note from Microsoft Regarding Office 2000 Installation

*“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:

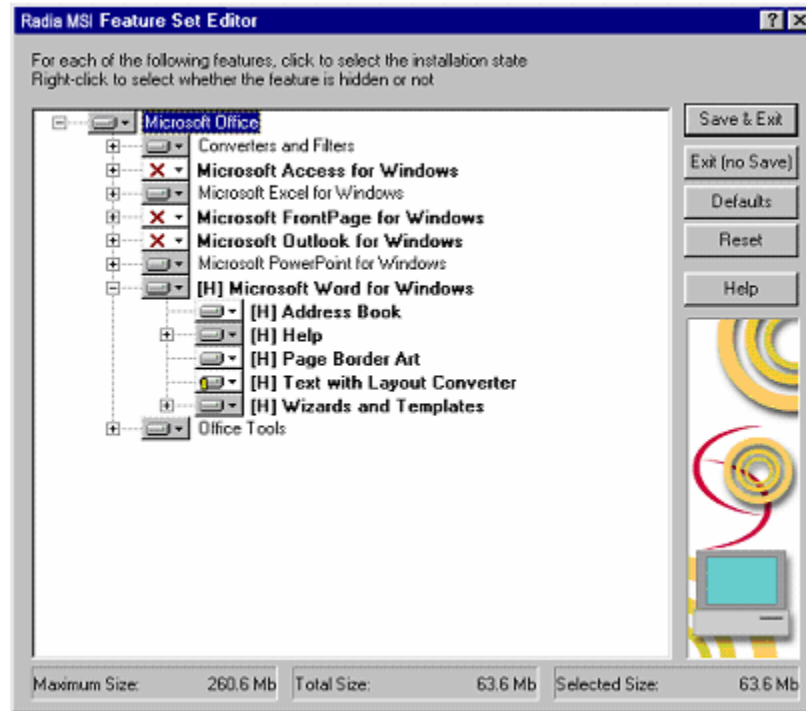


**Figure 90: The graphic settings for features.**

This can be seen in figure 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:

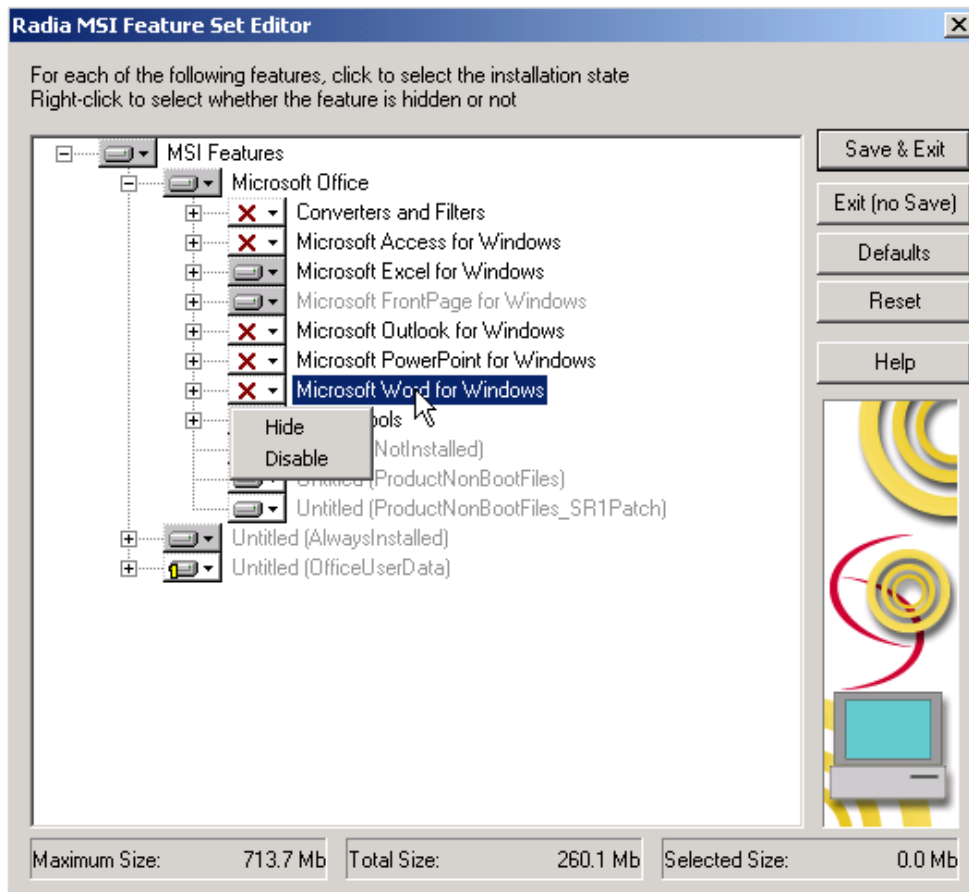


**Figure 91: Names of features appearing in boldface indicate a change to their attributes has occurred.**

In the figure 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:

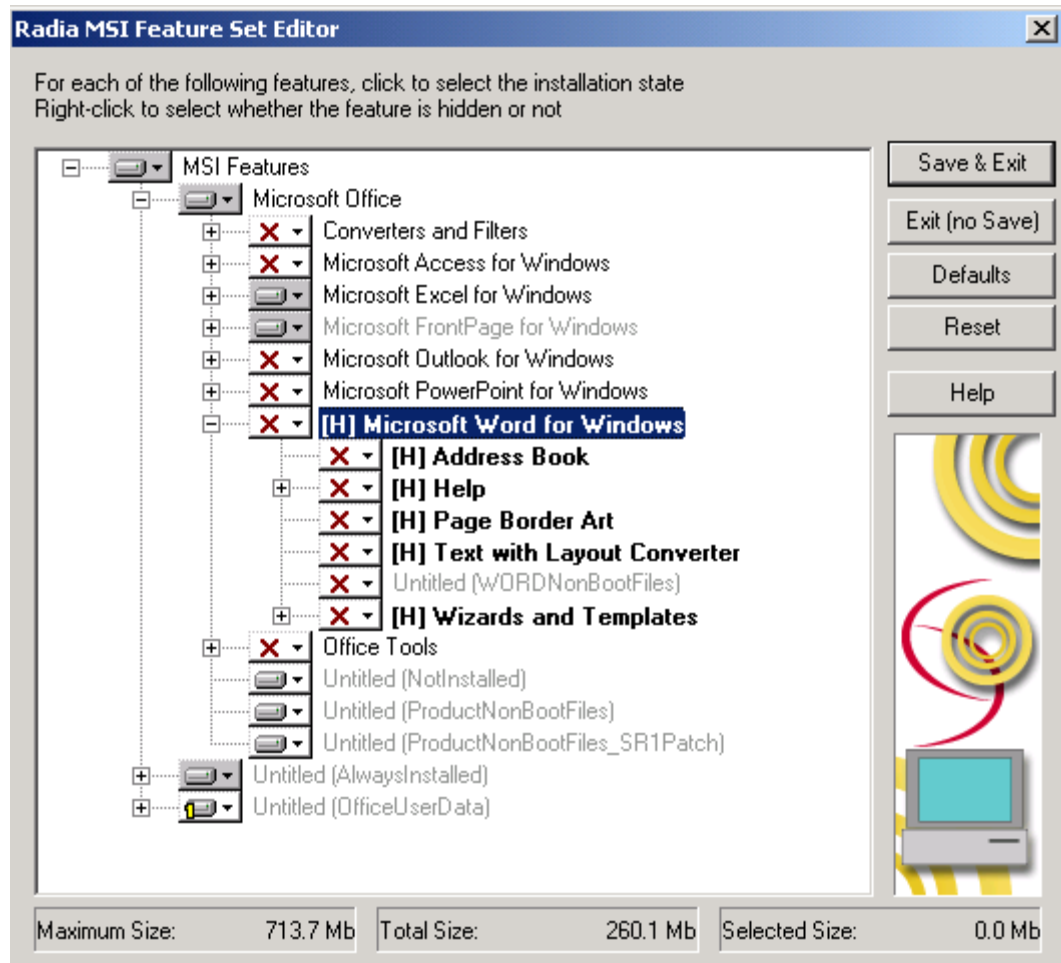


**Figure 92: A shortcut menu for Hide and Disable opens when a feature's name is right-clicked.**

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.

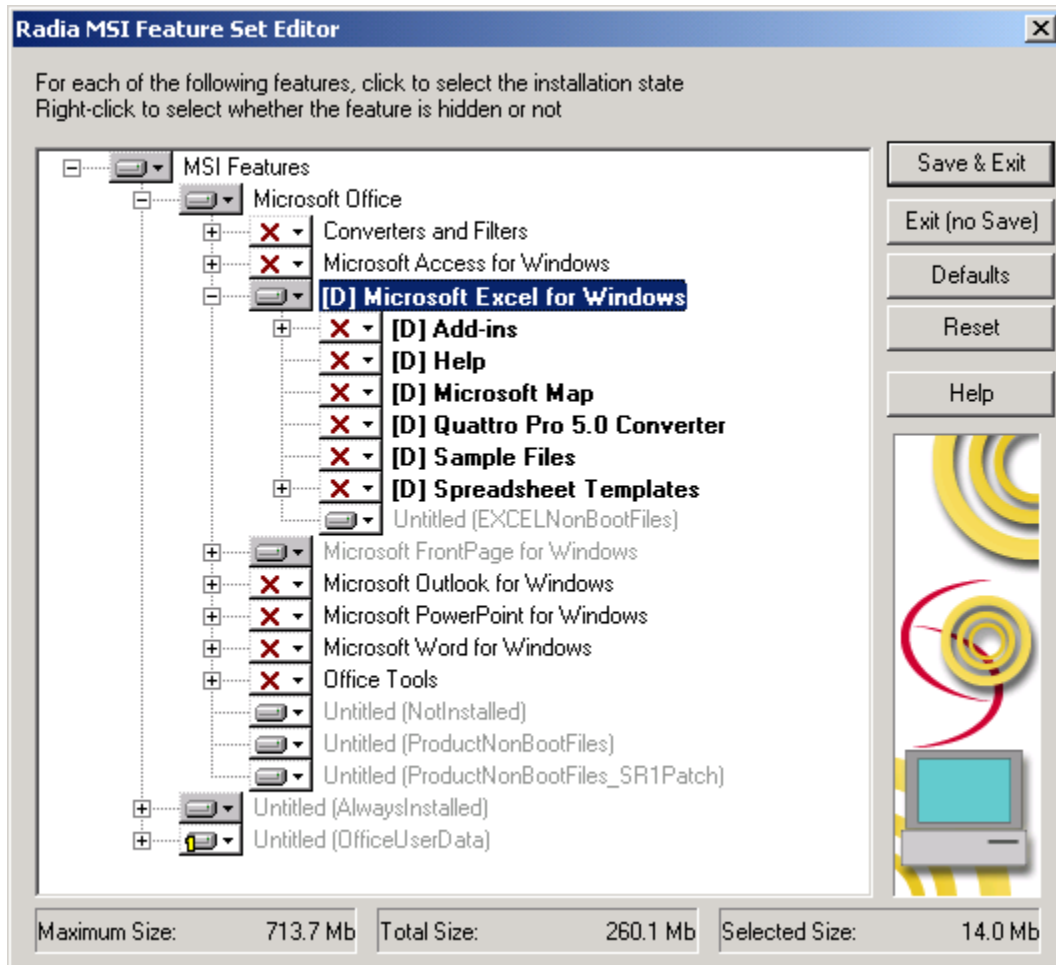
All features that are hidden appear in the Radia MSI Feature Set Editor tree with the prefix **(H)** before the name of the feature, as follows:



**Figure 93: 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:

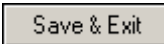
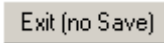
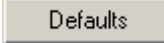

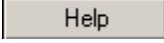


**Figure 94: All disabled features appear with the prefix (D) before their names.**

## Feature Set Editor Buttons

The Feature Set Editor buttons are located on the right side of the Feature Set Editor window. The following table describes the actions invoked by clicking these buttons.

**Table 29: Feature Set Editor Buttons**

Button	Description of Function
	Click <b>Save &amp; Exit</b> 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 <b>X</b> on the top right corner.
	Click <b>Exit (no Save)</b> to discard any changes that were made and to exit the Feature Set Editor. It is also possible to exit the Feature Set Editor by clicking on the <b>X</b> on the top right corner.
	Click <b>Defaults</b> 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 <b>Save &amp; Exit</b> .
	Click <b>Reset</b> 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 <b>Help</b> to access the 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

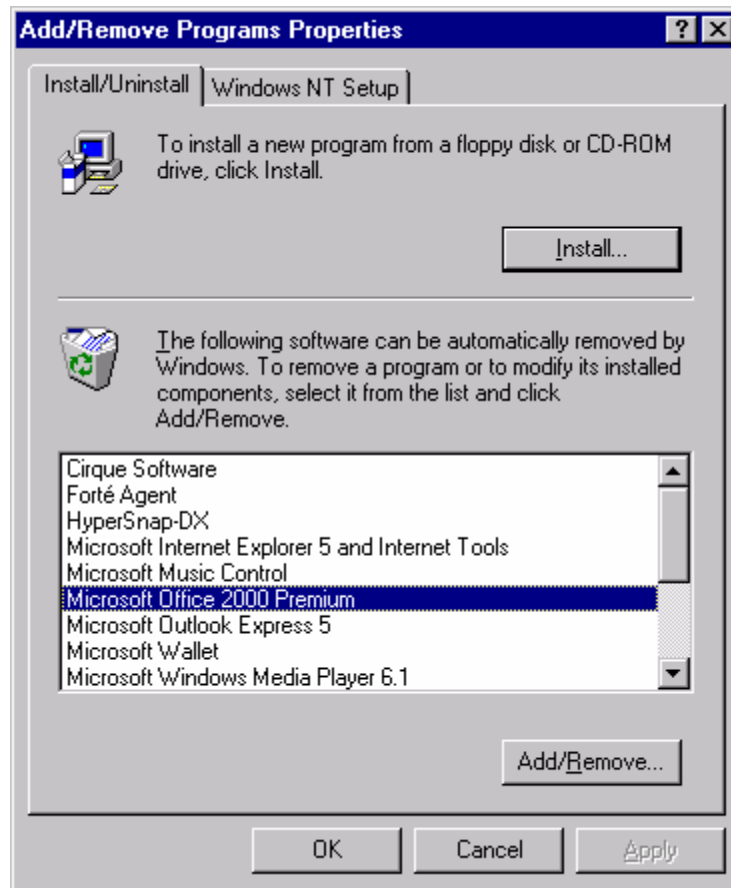
The number shown is the total number of bytes required on the client computer to install all features in the entire feature tree that have the **Run from My Computer** or **Run All from My Computer** attribute.

### 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, such as Office 2000, 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, as seen here:

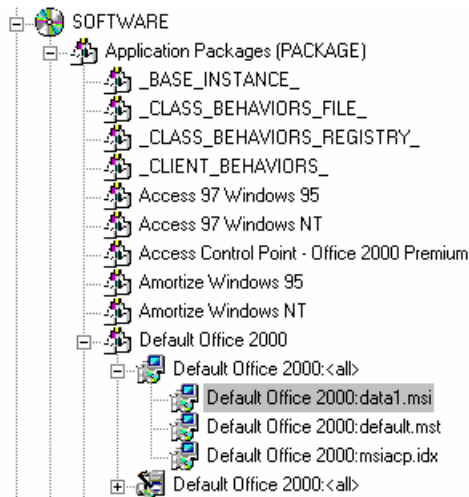


**Figure 95: The Add/Remove Programs Properties dialog box.**

Radia products give you the option to avoid listing Office 2000 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.

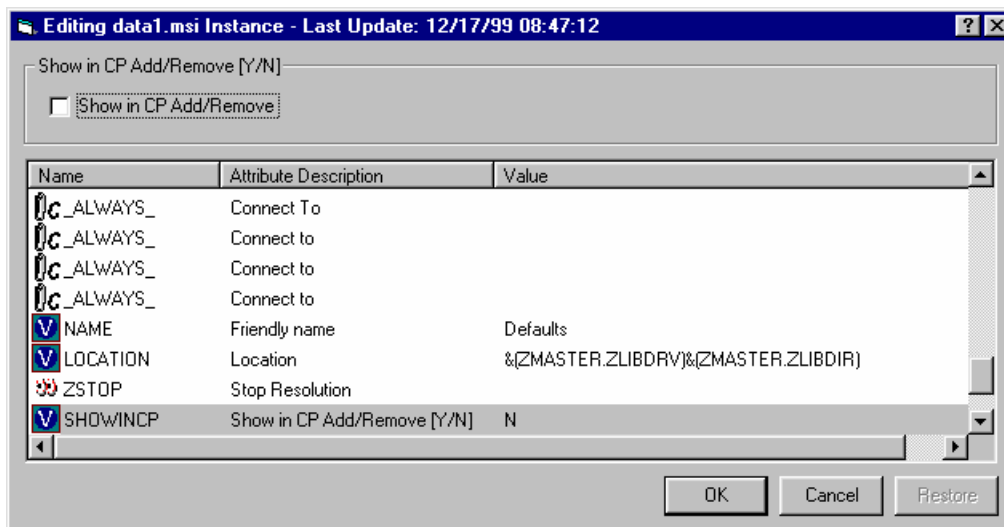
To completely avoid listing the application in the **Add/Remove Programs** applet, use System Explorer to change the setting of the SHOWINCP variable of the MSI class instance for the DATA1.MSI file that is connected to the Application Packages class instance:





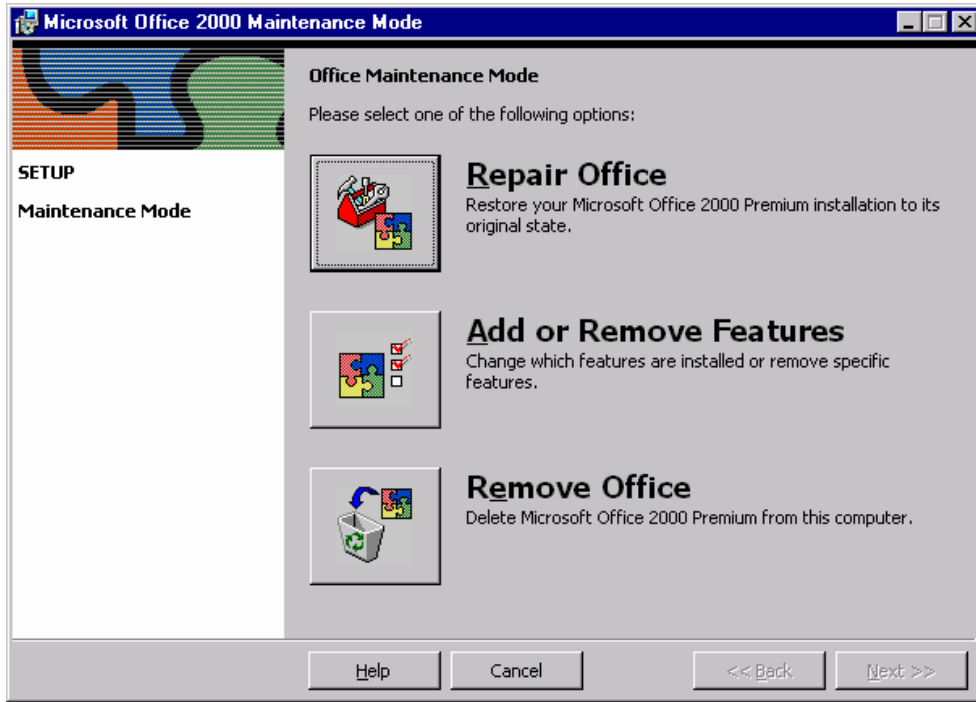
**Figure 96: The DATA1.MSI instance.**

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



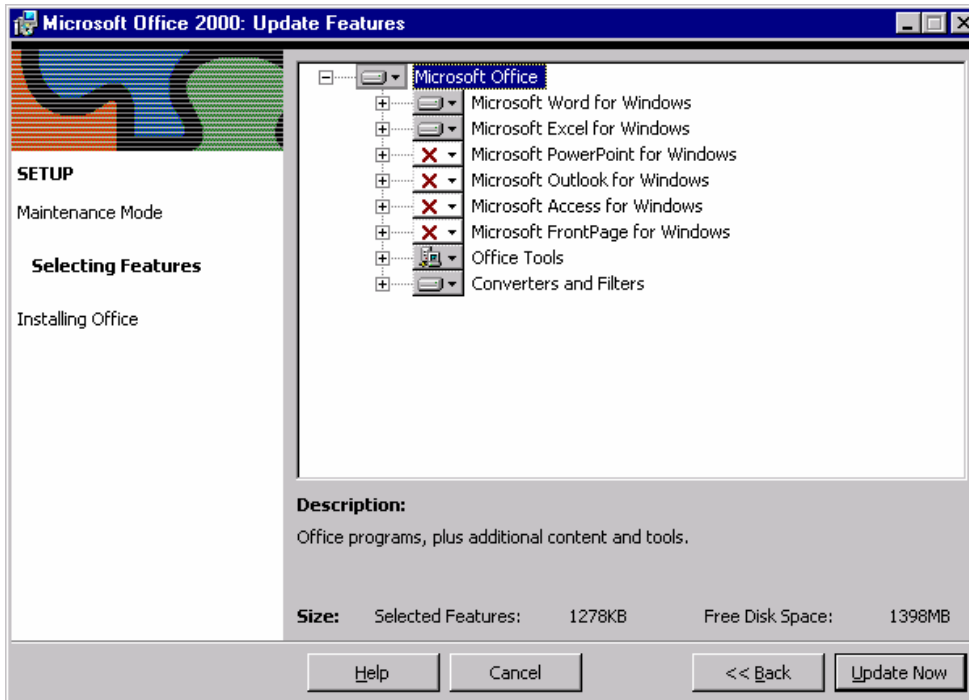
**Figure 97: The Editing DATA1.MSI Instance dialog box is used to change the value of SHOWINCP to N.**

When the end user selects Office 2000 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 Office installation on the client computer, as seen here:



**Figure 98: The Microsoft Office 2000 Maintenance Mode window.**

If the end user clicks **Add or Remove Features**, Microsoft Windows Installer displays its **Update Features** windows, as seen in the following figure.



**Figure 99: The Microsoft Office 2000: Update Features window.**

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

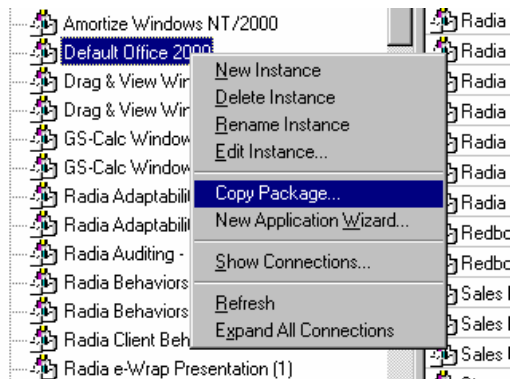
One of the key advantages of using Radia management products to manage Windows Installer-enabled applications is the ability to easily specify variants of the software package, and make those variants available to targeted populations of users.

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

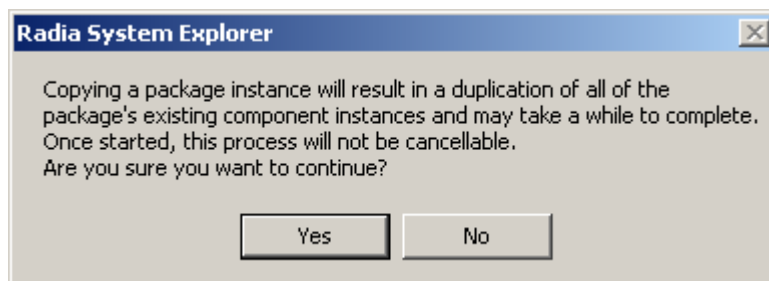
## To make a new variant set of features

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.



The following message opens.



- 2 Select **Yes** to continue the Copy Package function.

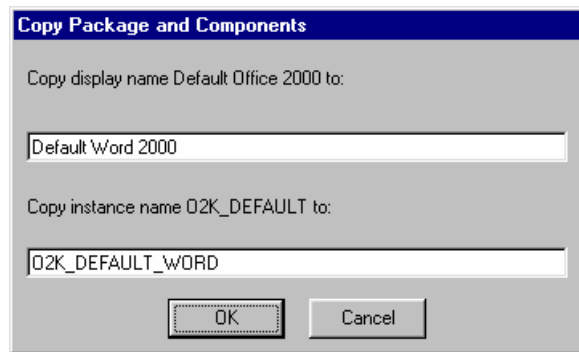
Select **No** to cancel the Copy Package function.

In this example, **Yes** was selected.

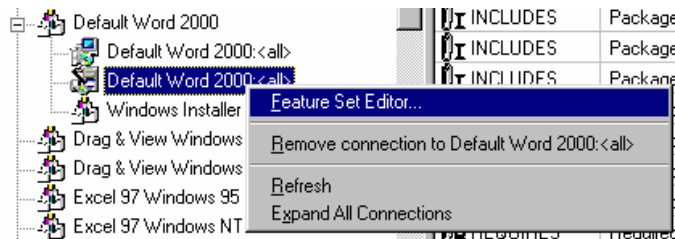
### Copying the Package

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:

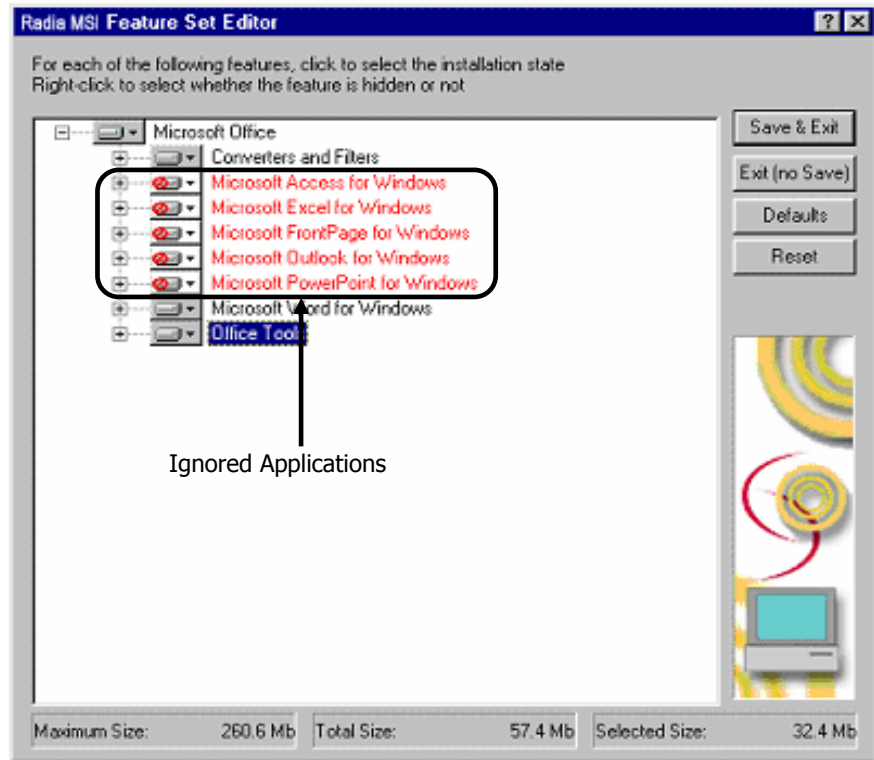


- 4 Click **OK** to continue. The Configuration Server takes a few moments to clone the package in the Radia database.
- 5 When the newly cloned package appears in the System Explorer's tree view, right-click on the MSIFEATS instance beneath the package in the tree view, and select **Feature Set Editor** from the shortcut menu:



The Radia MSI Feature Set Editor opens.

- In this example, edit the package to install and manage Word 2000 with the default settings from the Office 2000 CD-ROM. None of the other Office 2000 applications will be deployed by this package. Set the properties of the applications that are not to be deployed to **Ignore**:



- 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 Word 2000 as a stand-alone application.

## Managing Feature Sets in Native Mode

In some enterprises, packaging applications and administering Radia are separate functions. You might 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 System Explorer to remove the connection to SOFTWARE.MSIFEATURES.packageprefix\_\* from the Application Packages (PACKAGE) instance.
- 3 If you have multiple MST files, use the 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.



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 MSIFEATURES 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 update and repair.



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), the new feature from the MST file will be included in the MSI service. However, if there is a feature in the MST file that already exists in the MSIFEATS object, the feature state indicated in MSIFEATS will be used, and the one in the additional MST file will be disregarded. If you want to manage the MSI package using only your MST file, use the procedure 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.



---

## 9 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.

# The Registry File Editor

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

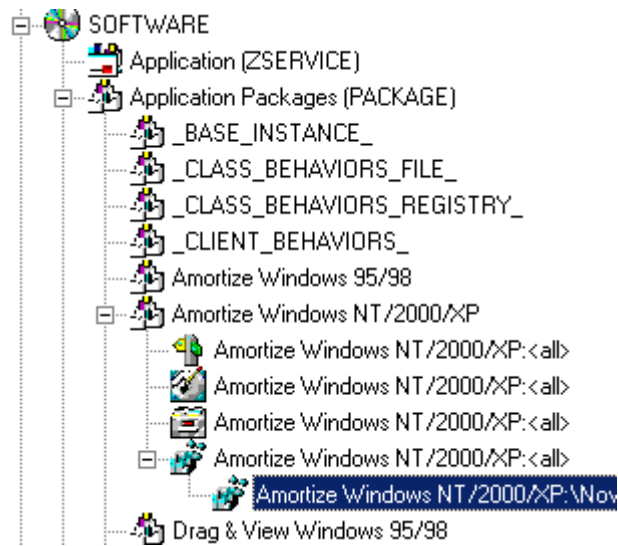
When using HP OpenView Packager Using Radia (Packager) to create a package, all of the registry changes for that package are stored in a file (with an `.EDR` extension) in the `LIB` directory. The Registry File Editor allows you to modify this file. Then, the file is sent (or re-promoted) to the Configuration Server and the instance in the Radia database is updated.

See the *Installation and Configuration Guide for the HP OpenView Application Manager Using Radia (Application Manager Guide)* for information on creating and promoting packages.

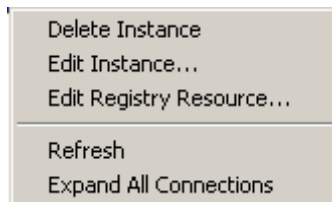
## Accessing the Registry File Editor

### To access the Registry File Editor

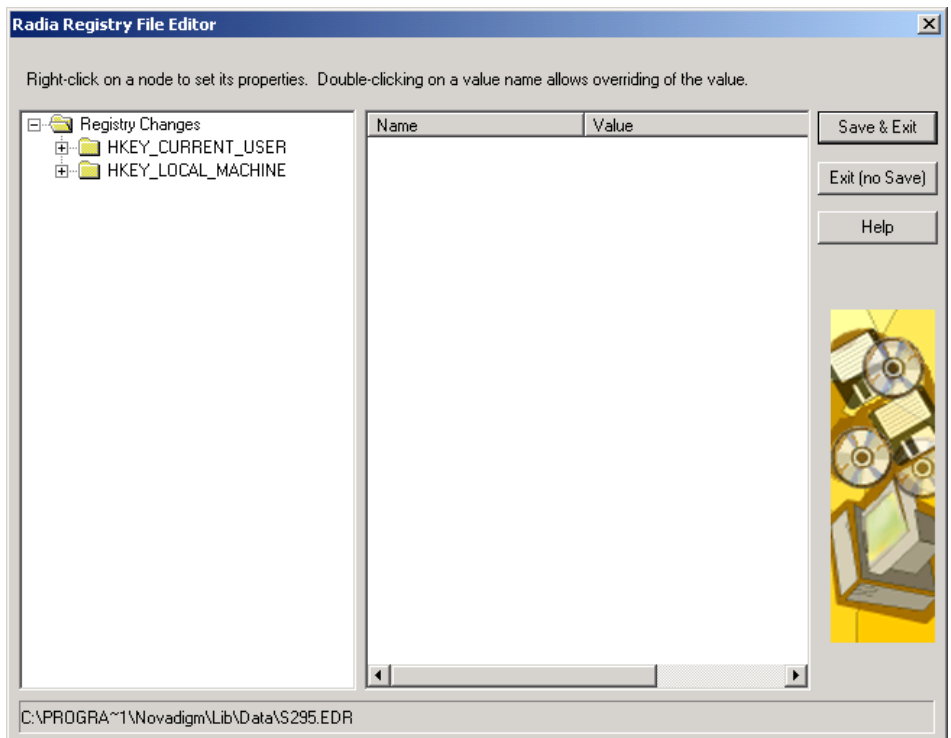
- 1 Locate the **Registry Resource** instance that represents the software package whose registry settings you wish to change in the Radia database.
- 2 Right-click the instance in the tree view. A context menu opens. For example, locate the registry instance for the Amortize Windows package and right-click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



3 Select **Edit Registry Resource**.



The Registry File Editor dialog box opens.



- The **tree view** (on the left) displays nodes that represent keys in the registry file.
- The **list view** or **value name** (in the middle) displays the contents of a selected registry key.
- The **buttons** (on the right) allow you to Save & Exit, Exit without saving, or access the Help function in the Registry File Editor.

## Expanding and Collapsing the Tree

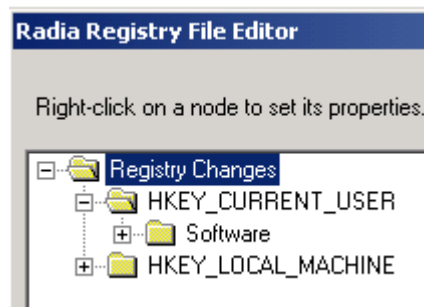
Nodes are grouped into a hierarchy. When you set a property to a node, the Registry File Editor applies changes to all nodes within that node's branch in the tree view.

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.

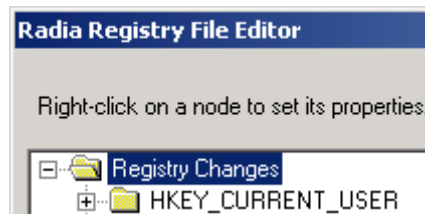
The plus sign changes to a minus sign and the next level opens beneath the selected node in the tree view.



### To collapse a single node on the tree

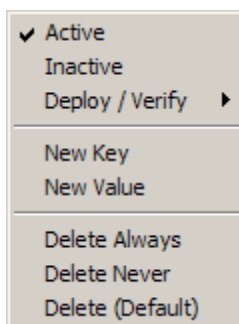
- Click the minus sign in the small box next to the node.

The minus sign changes back to a plus sign.




## Set Properties Context Menu

To set properties for the Registry File Editor, right-click on any node of the tree view in the Registry File Editor. A context menu opens.



For specific information regarding the options available through this context menu, see the following table.

**Table 30: Registry File Editor Context Menu Options**

Action	Description
Active	Select <b>Active</b> to include the registry keys in the package.
Inactive	Select <b>Inactive</b> to prevent the registry keys from being included in the package. Inactive nodes appear with a  symbol covering its folder.
Deploy/Verify	Click <b>Deploy/Verify</b> to access the following submenu: <ul style="list-style-type: none"><li>• <b>Unconditional/Enforced</b> Radia deploys the registry key every time the subscriber connects to the Configuration Server, even if it already exists on his desktop. This setting repairs the registry key if it is modified or deleted. Use this setting for registry keys that must remain invariant in order for the application software to function properly.</li><li>• <b>Unconditional/Exist</b> 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.</li><li>• <b>Exist/Exist</b> Radia deploys the registry key only if it doesn't exist on the subscriber's desktop, during both the initial deployment and on</li></ul>



Action	Description
	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 <b>New Key</b> if you would like to add a new value to your registry. Clicking <b>New Key</b> will display a dialog box to enter the <b>New Key</b> name. Click <b>OK</b> to accept the new key, or <b>Cancel</b> to return to the Registry File Editor.
New Value	<p>Click <b>New Value</b> to add a value to a newly created key or to add a value to an existing registry key. Clicking <b>New Value</b> opens a dialog box where you will be required to enter the following information:</p> <p><b>Value Name</b> 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 Registry File Editor window.</p> <p><b>Data Type</b> 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"> <li>• <b>REG_SZ</b> A text sequence most typically used for descriptive information. This is the default setting.</li> <li>• <b>REG_DWORD</b> Is data represented by an integer that is 4 bytes long and is displayed in the binary, hexadecimal, or decimal format.</li> <li>• <b>REG_EXPAND_SZ</b> Is an expandable data string. This text contains a variable that is replaced when called by an application.</li> </ul> <p><b>REG_MULTI_SZ</b> A string containing multiple text entries of user readable text.</p>
Delete Always	Click <b>Delete Always</b> if you would like all values under a Registry key to be deleted during a Radreg32 Delete operation. Note that when this option is turned on, all values under the key are deleted <i>without</i> restoring their original registry values in the backup EDR file.
Delete Never	Click <b>Delete Never</b> if you would like to retain all values under a key during a Radreg32 Delete operation.
Delete (Default)	Click <b>Delete (Default)</b> if you would like the Radreg32 Delete operation to exhibit normal Delete behavior with regards to deleting or restoring

Action	Description
	values.

## Editing a Registry Key

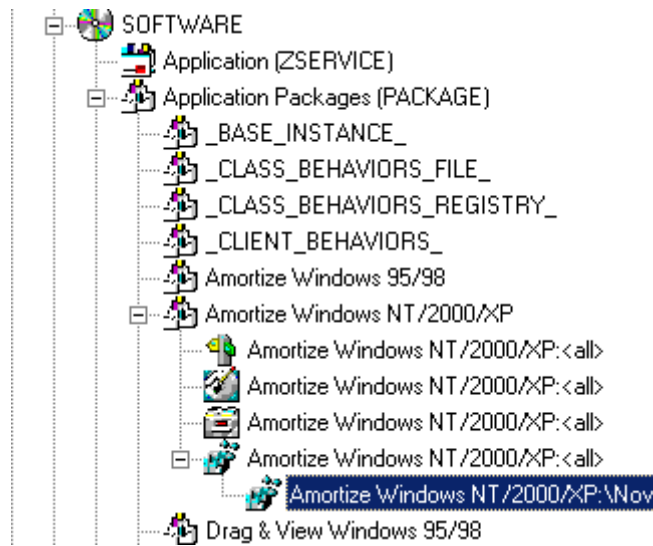
Editing a registry key's value is accomplished by accessing the key through the Registry File Editor. Follow the example provided in this section to edit the information in a registry key.



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

### To edit the data in a registry key

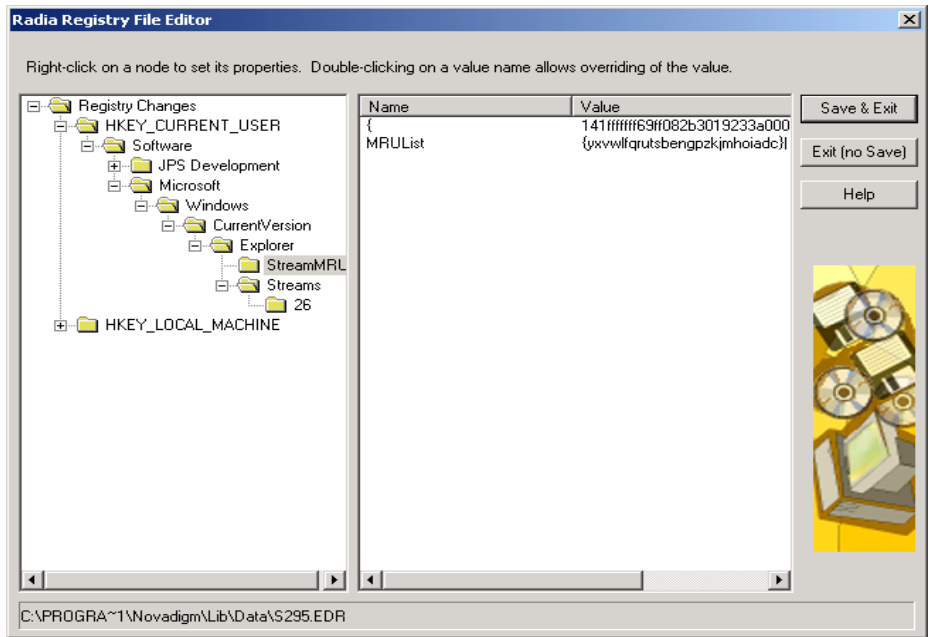
- 1 From the Windows Taskbar, click **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**. The Radia System Explorer Security Information dialog box appears.
  - ▶ The **User ID**, as shipped from HP, is RAD\_MAST. No password is necessary. This might have been changed 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 package **Amortize Windows NT/2000/XP**.
- 6 Expand the **Registry** instance.



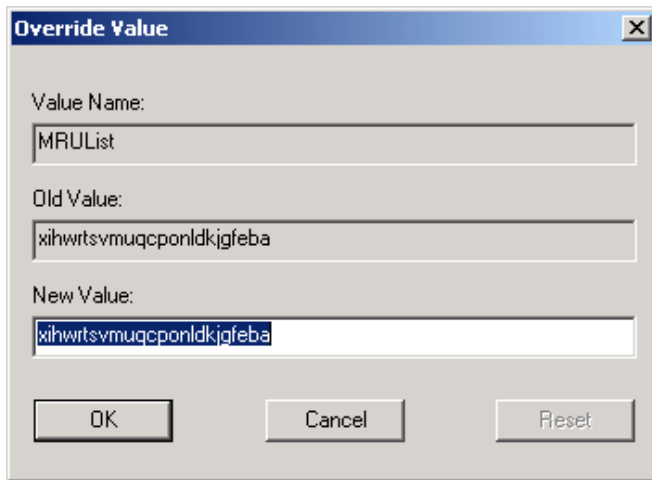
- 7 Right-click the registry file and select **Edit Registry Resource** from the context menu.

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.



The Override Value dialog box opens.



10 Edit the data in the **New Value** text box.

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

or

Click **Cancel** to close the dialog box without saving the changes.

### To restore the data in a registry key

1 Double-click the name of the registry key in the list view.

2 Click **Reset**.

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

or

Click **Cancel** to close the dialog box without saving the changes.

## Closing the Registry File Editor

Close the Registry File Editor when you are finished making changes to instances of the Registry file.

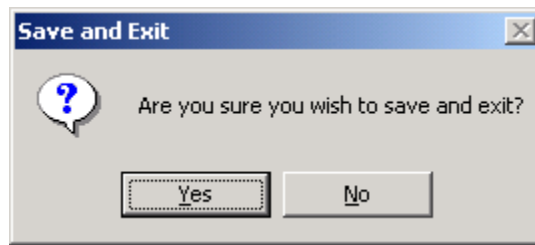
### To close the Registry File Editor

1 Click **Save & Exit** to save the changes and close the dialog box.

or

Click **Exit (no Save)** to discard the changes and close the dialog box.

2 Click **Yes** to confirm that you want to save the changes and close the dialog box.

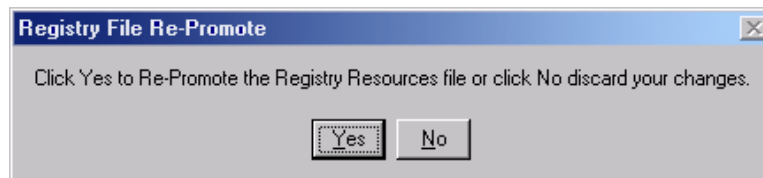


or

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

- 3 Click **Yes** to re-promote the Registry Resources file to the Radia database.

The System Explorer saves the changes to the file and sends it to the Configuration Server to update the instance in the Radia database.



or

Click **No** to discard your changes and return to the System Explorer.

## 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.





---

# 10 Support for the Inventory Manager

At the end of this chapter, you will:

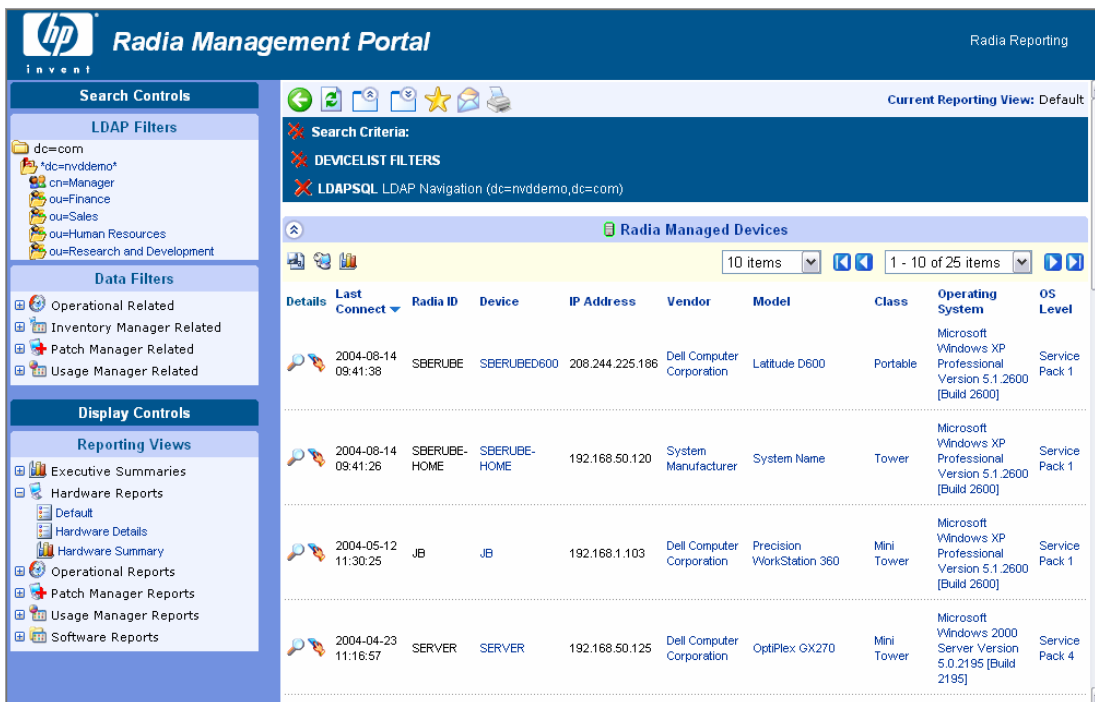
- Understand the role of the Inventory Manager client in discovering configuration information on remote computers.
- Be familiar with the Radia components used to collect, report and view the inventory discovery results.
- Understand Web-based Enterprise Management (WbEM) and the Radia client.

## About the Inventory Manager

The **Inventory Manager Client** of the HP-OpenView Inventory Manager Using Radia (Inventory Manager) discovers configuration information on remote computers. It enables centralized reporting and administration based upon the discovery results.

The Inventory Manager Client can be used with the Inventory Manager Server or the HP-OpenView Messaging Server Using Radia (Messaging Server) to report the discovery data to an ODBC-compliant database. If the Inventory Manager Server is used, it stores and maintains discovery information to be viewed within the database. If the Messaging Server is used, it posts the client discovered information directly to the database.

The preferred method for viewing the reported data is to use the Reporting Server. The Reporting Server interface provides a dynamic and intuitive way to use Radia SQL data for reporting and overall environmental assessment.



**Figure 100: Reporting Server Web interface supports Inventory Manager auditing.**

## Overview of Auditing

The Inventory Manager allows for three types of audits: file auditing, WBEM auditing, and Hardware auditing. The AUDIT domain of the Radia Database contains the classes required to configure the tasks needed to collect the inventory information and manage the client computer's assets.

Systems administrators can use the System Explorer, which is available for 32-bit Windows platforms, or the Management Portal, available for UNIX and Win32 platforms, to manipulate the contents of the Radia Database for inventory auditing. For example, using these tools System administrators can specify what inventory management tasks to perform and on which client computers to perform them.

The collection of inventory information occurs on the Inventory Manager Client computer when a subscriber connects to the Configuration Server and installs the auditing software 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 logon script.
  - when the user double-clicks a desktop Radia Connect icon.
  - according to a schedule.
  - using the Notify capabilities of the Configuration Server.

The results of the inspection are then sent back to the Configuration Server. Any unwanted files discovered on a user's computer can be captured or deleted. This enables administrators to remove unauthorized content, such as games, from end users' computers.

The Configuration Server can store these inspection results in the Radia Database PROFILE file or can use the Messaging Server to insert the results into an ODBC-compliant database.

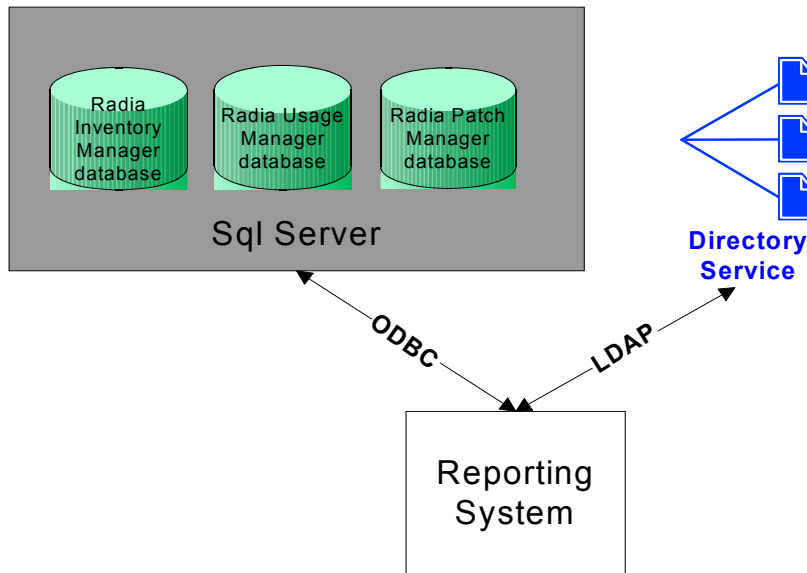
The Reporting Server interface provides a dynamic and intuitive way to use the collect Radia data for reporting and overall environmental assessment.

For more information on auditing using the Inventory Manager, refer to the *Installation and Configuration Guide for the HP-OpenView Inventory Manager Using Radia*.

## About the Reporting Server

As part of the Radia extended infrastructure, the web-based Reporting Server allows you to query the combined data in the databases of the Inventory Manager, HP OpenView Patch Manager Using Radia (Patch Manager), and HP OpenView Usage Manager Using Radia (Usage Manager), and create detailed reports. In addition, you have the option of mounting an existing

LDAP directory, which allows you to filter your data using your LDAP directory levels.



**Figure 101: Radia Reporting Environment**

The Reporting Server interface provides a dynamic and intuitive way to use Radia SQL data for reporting and overall environmental assessment.

See the *Installation and Configuration Guide for the HP-OpenView Reporting Server Using Radia* for more information on how to install the Reporting Server and create a Radia Reporting environment for your SQL databases for Radia, such as inventory, patch, and usage databases, as well as an optional LDAP directory.

## Radia and WbEM

The Inventory Management client queries the Web-based Enterprise Management (WbEM) namespace (that is, the WbEM database) and sends the results back to the Configuration Server. All information collected by

WbEM is available to the Inventory Manager client. The collected information is then stored in the Inventory ODBC database.

For client computers with WbEM installed, the Inventory Manager executes an HP-proprietary method (RIMWBEM) to query the WBEM namespace.

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

## CIM Schema and Inventory Collection

As a guide for collecting hardware and software inventory, HP uses the Common Information Model (CIM) schema version 2.6. This allows inventory to be collected based on industry standards, as defined by the Distributed Management Task Force (DMTF).

The CIM schema allows real-world objects to be mapped to objects defined in the different schema classes and attributes. After data is discovered using these standards, the output is collected by Radia and is available for reporting purposes.

## Summary

- The 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 Inventory Manager.
- The Inventory Manager queries the WbEM namespace and sends the results back to the Configuration Server.
- Complete information concerning the Inventory Manager, the AUDIT domain and the services provided through Radia, can be found on the HP OpenView web site.





# A Using System Explorer Help

## Getting Help

The System Explorer offers two kinds of help:

- Traditional Microsoft Windows-style help  
and
- Context-sensitive help


### Traditional Microsoft Windows-style Help


Traditional Microsoft Windows-style help is available via the **Help** menu and via a **Help** button on the Radia MSI Feature Set Editor dialog box. See Table 6 on page 38 for the options available from the Help menu, and see About the Radia MSI Feature Set Editor starting on page 249 for information on the Feature Set Editor help.

### Context-Sensitive Help

Context-sensitive help is available for almost all controls in the System Explorer user interface. The following buttons provide access to context-sensitive help:

**Table 31: Buttons to Access the System Explorer Help**

Button	Button Location
	Upper right corner of most windows and dialog boxes

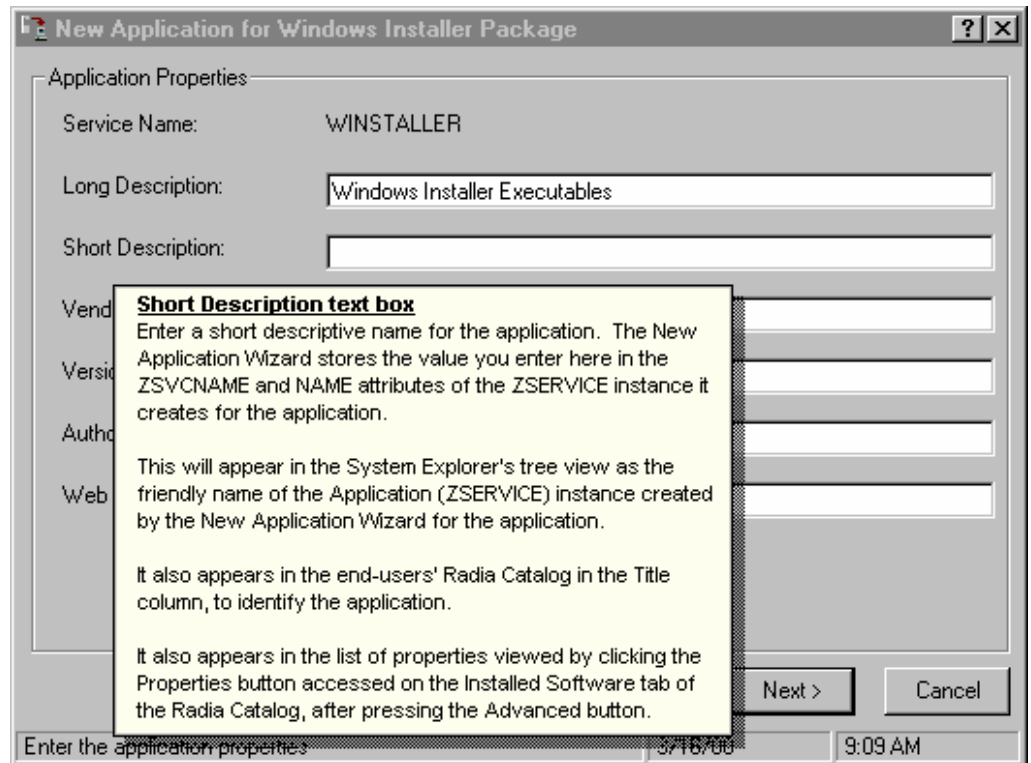
Button	Button Location
	System Explorer window toolbar

## To invoke context-sensitive help

- 1 Click one of the two buttons in the table above.

The mouse cursor changes shape (to a pointer with a question mark) to indicate that you are about to seek context-sensitive help.

- 2 Click this cursor on a control or area of a window for which you'd like help. An information box describing the control or window area you selected will appear:

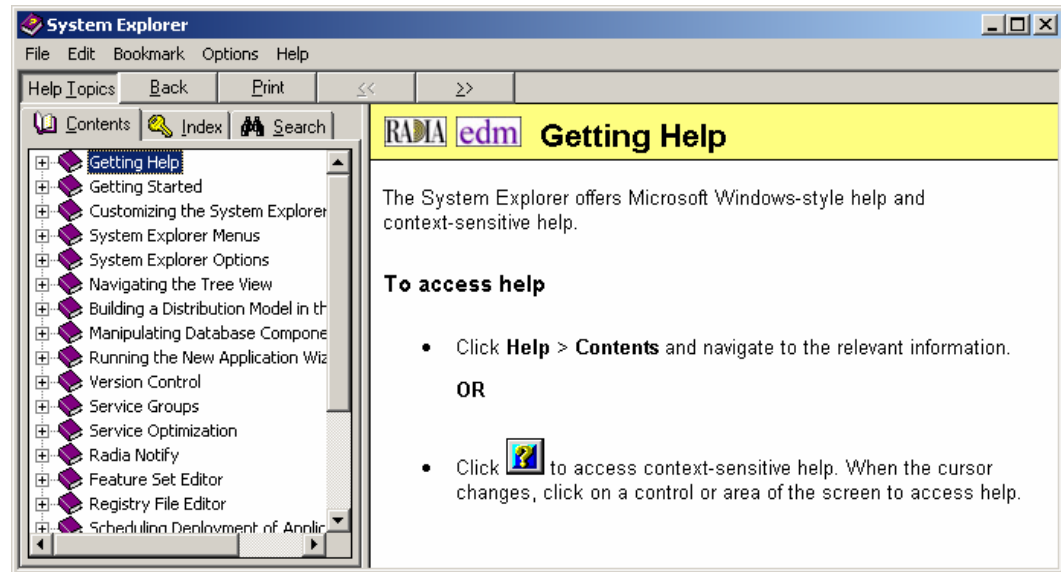


The information box automatically closes when you click again.

or

- 3 Highlight the field you need help on.
- 4 Press the **F1** function key.

Help that is appropriate to the currently selected control (i.e., the control with input focus) appears in a pop-up information box.

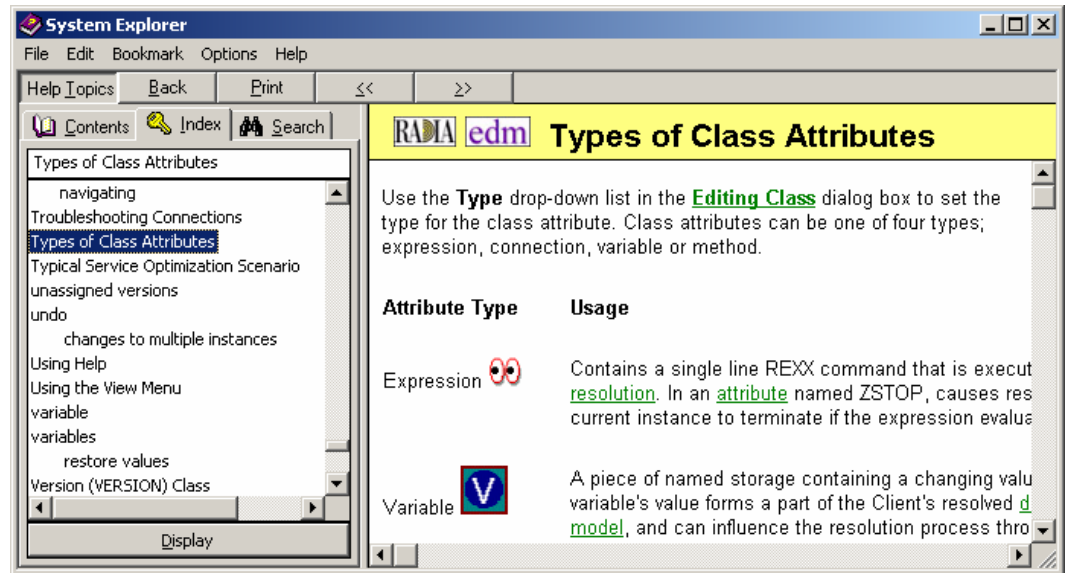


The help system interface consists of a window with two panes. The left pane contains controls for finding the help information you want, which is displayed in the list view. It has three tabs that offer different ways for navigating the help information.

- **Contents** tab
- **Index** tab
- **Search** tab

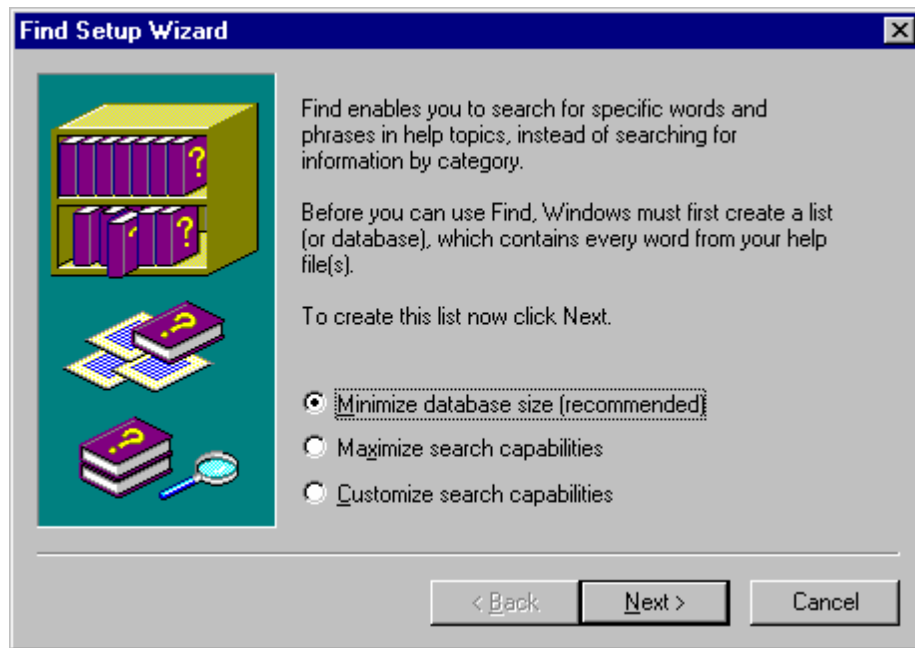
**Contents** tab uses a book and page metaphor in a tree control. The help information is organized in topic *books*, which have *sub-topic pages*. Open a book by clicking on the plus sign next to the book icon of the topic you want to inspect. Display a page in the list view, by clicking on its title in the left pane.

**Index** tab presents an alphabetic index of available pages, similar to the following:



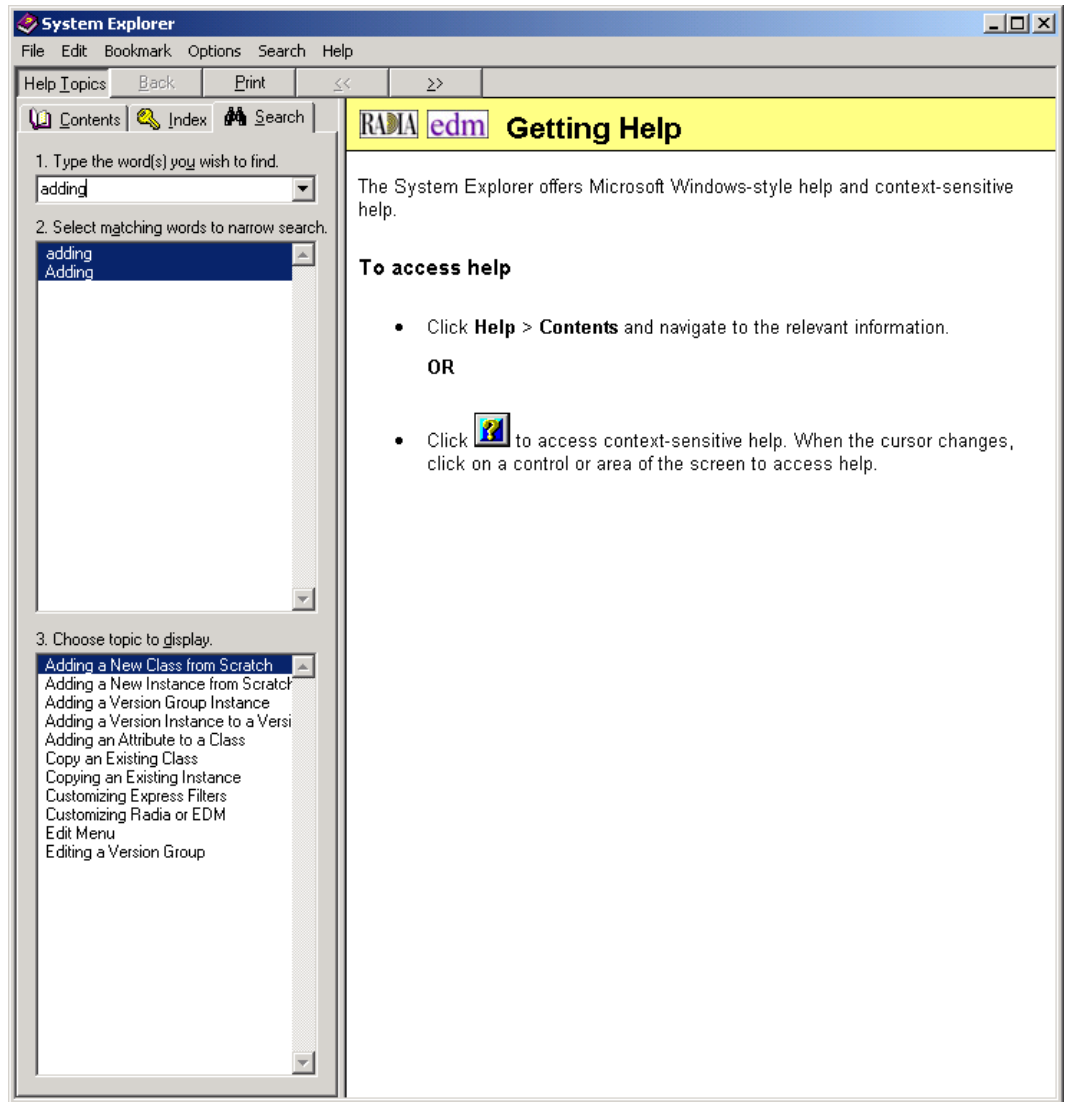
Click on the index entry in the left pane to display its page of information in the list view.

**Search** tab enables free-text search of the help information. The first time you click the **Find** tab, the help system builds an index of the help information. It presents the following dialog box in which you can indicate which indexing level you prefer:



After the index is built (which takes a few seconds), the **Search** control appears in the left pane.

Enter the text you want to search for (such as “adding”), and click on the topic you want to display:

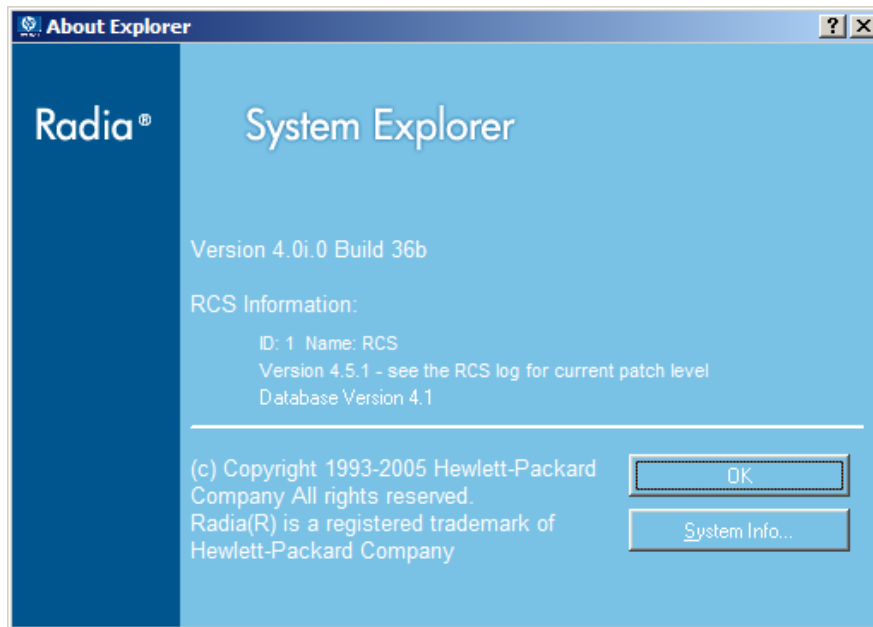


The text you searched for is highlighted in the page displayed in the list view.

The Help system describes features in both the System Explorer for HP OpenView Using Radia as well as the System Explorer for EDM (a predecessor to Radia). Certain features of the System Explorer apply exclusively to Radia and others exclusively to EDM. The

non-scrolling region at the top of the list view contains a logo indicating if the feature described applies to the Radia System Explorer, the EDM System Explorer, or both.

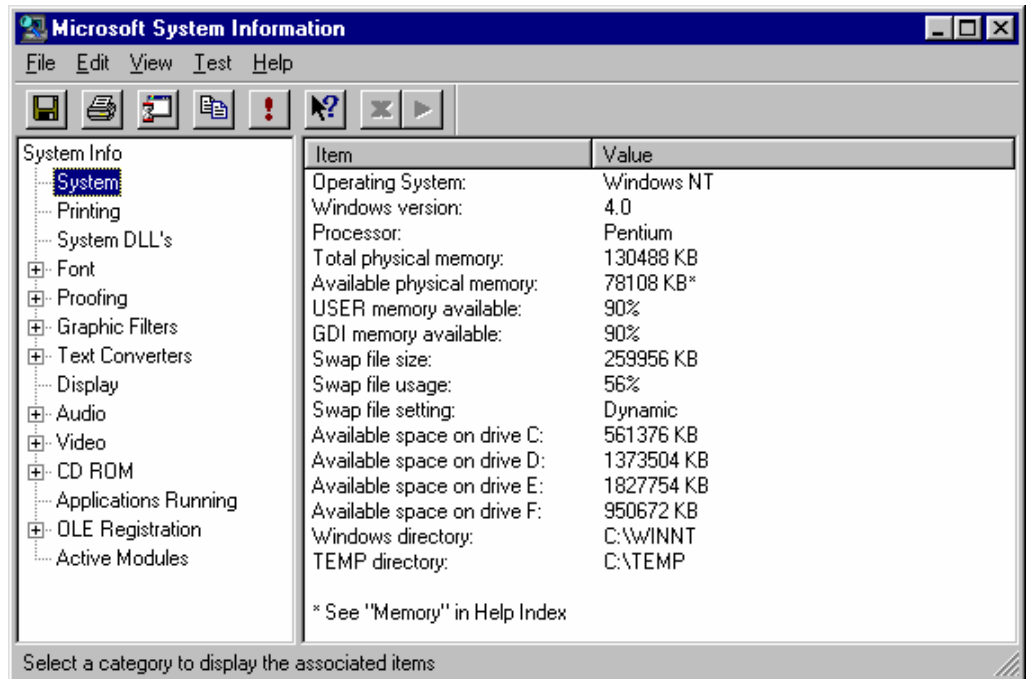
- The **Search for Help on** option has the same functionality and appearance as the **Contents**. See this section for assistance accessing and navigating through the help topics.
- The **HP on the Web** choice on the **Help** menu opens a sub-menu with links to the HP OpenView web site's main home page and Technical Support home page. Selecting one of these links opens your default Web browser and connects with the selected page, over the Internet.
- The **About Explorer** choice on the **Help** menu displays the following dialog box:



The Help → About Explorer window displays the Radia client **Version** and **Build** number, followed by the **ID**, **Name**, and **Version** of the Configuration Server.

- Click **OK** to exit the About Explorer dialog box.

- Click **System Info** to open the standard Microsoft System Information dialog box.



This application analyzes your machine and presents the results in an easy-to-navigate format. The information can be useful when you are troubleshooting problems on your computer.



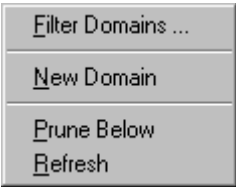

## B System Explorer Shortcut Menus


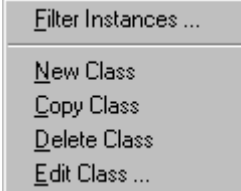
In the System Explorer you can add, copy, edit, rename, and delete Radia database components (domains, classes, and instances). It also provides extensive support for making correct connections between instances.

You can access these functions from shortcut menus that open when you right-click targets in the 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.

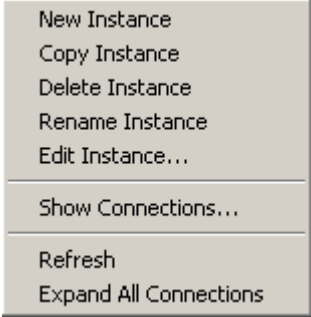
- ▶ The **Prune Below** menu choice appears only when the tree has been opened below the target of the right-click.

**Table 32: Primary, Domain, and Class Shortcut Menus**

Target of Right-Click	Context Menu
PRIMARY file in the tree view	
Any domain in the tree view	

<b>Target of Right-Click</b>	<b>Context Menu</b>
Any class in the tree view	
Any class in the list view of the System Explorer window	

**Table 33: ADMIN Instance Context Menus**

<b>Instance Name</b>	<b>Context Menu</b>
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)	

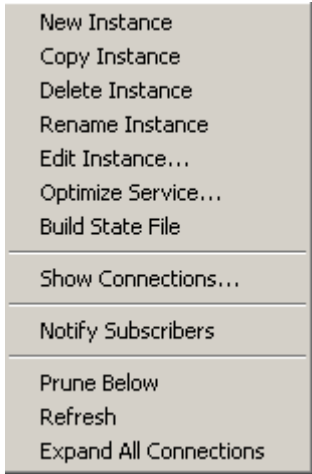
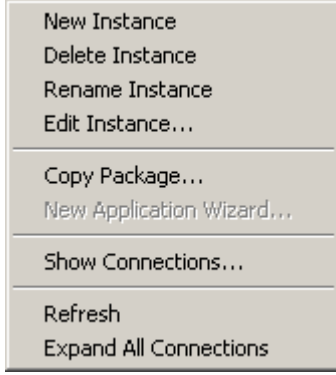
Instance Name	Context Menu
Attribute Verify (ZVERA) Console Privileges (ZCONUSER) Name Lists (32) (ZLIST32) Name Lists (8) (ZLIST)	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div>

**Table 34: AUDIT Instance Context Menus**

Instance Name	Context Menu
Application (ZSERVICE)	<div style="border: 1px solid black; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> <li>Optimize Service...</li> <li>Build State File</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Show Connections...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Notify Subscribers</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Prune Below</li> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div> <p>Note: Build State File is available if the Radia Extensions for Windows Installer is installed.</p>

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

**Table 35: NOVADIGM Instance Context Menus**

Instance Name	Context Menu
Application (ZSERVICE)	 <p>Note: Build State File is available if the Radia Extensions for Windows Installer is installed.</p>
Application Packages (PACKAGES)	

Instance Name	Context Menu
Desktop (DESKTOP)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Show Connections...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div>
File Resources (FILE) Path (PATH) Unix File Resources (UNIXFILE)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div>
Maintenance Location (MAINTLOC)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div>

**Table 36: PATCH Instance Context Menu**

Instance Name	Context Menu
Patch (PATCH)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul> </div>

**Table 37: POLICY Instance Context Menus**

Instance Name	Context Menu
Countries (COUNTRY) Mobile Device Config (MBLCONFIG) Multicast (MULTICAST) PDACONFIG (PDACONFIG) Server Stagers (STAGER)	<div style="border: 1px solid gray; padding: 5px;">             New Instance              Copy Instance              Delete Instance              Rename Instance              Edit Instance...           </div> <hr/> <div style="border: 1px solid gray; padding: 5px;">             Refresh              Expand All Connections           </div>
Departments (DEPT) Workgroups (WORKGRP)	<div style="border: 1px solid gray; padding: 5px;">             New Instance              Copy Instance              Delete Instance              Rename Instance              Edit Instance...           </div> <hr/> <div style="border: 1px solid gray; padding: 5px;">             Show Connections...           </div> <hr/> <div style="border: 1px solid gray; padding: 5px;">             Refresh              Expand All Connections           </div>
Users (USER)	<div style="border: 1px solid gray; padding: 5px;">             New Instance              Copy Instance              Delete Instance              Rename Instance              Edit Instance...              Other Services...           </div> <hr/> <div style="border: 1px solid gray; padding: 5px;">             Show Connections...           </div> <hr/> <div style="border: 1px solid gray; padding: 5px;">             Refresh              Expand All Connections           </div>

**Table 38: SOFTWARE Instance Context Menus**

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



Instance Name	Context Menu
Behavior Services (BEHAVIOR)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Show Connections...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul>
Class Defaults (METACLAS) Desktop (DESKTOP) Dialog Services (DIALOG)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Show Connections...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul>
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)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Refresh</li> <li>Expand All Connections</li> </ul>

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

**Table 39: SYSTEM Instance Context Menus**

Instance Name	Context Menu
Application Manager (ZCOMMAND)	<ul style="list-style-type: none"> <li>Execute Command</li> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> <li>Refresh</li> <li>Expand All Connections</li> </ul>
Consoles (ZCONSOLE) Methods (ZMETHOD) Radia Processes (PROCESS) SQLTABLE (SQLTABLE) System Files (ZFILE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Instance</li> <li>Rename Instance</li> <li>Edit Instance...</li> <li>Refresh</li> <li>Expand All Connections</li> </ul>

**Table 40: ADMIN Domain Instance Context Menus**

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

**Table 41: AUDIT Domain Instance Context Menus**

Class Name	Context Menu
Application (ZSERVICE)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Notify Subscribers</li> </ul> </div>
Audit Packages (PACKAGE)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> <li>Add Components...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Copy Package...</li> <li>New Application Wizard...</li> </ul> </div>
Behavior Service (BEHAVIOR) Desktop (DESKTOP) File (FILE) File Scanner (FILESCAN) Path (PATH) WBEM (WBEM)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> </div>
Client Methods (CMETHOD) Inventory Options (RIMOPTS) Inventory Scanners (SCANNER) Scheduling (TIMER) Virtual Mgr Location (MGRVLOC)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> </div>
File Scanner Filters (FILTER)	<div style="border: 1px solid gray; padding: 5px;"> <ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> </div>

**Table 42: NOVADIGM Instance Context Menus**

Class Name	Context Menu
Application (ZSERVICE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Notify Subscribers</li> </ul>
Application Packages (PACKAGE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Copy Package...</li> <li>New Application Wizard...</li> </ul>
Desktop (DESKTOP)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul>
File Resources (FILE) Path (PATH) Unix File Resources (UNIXFILE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul>

**Table 43: PATCH Instance Context Menus**

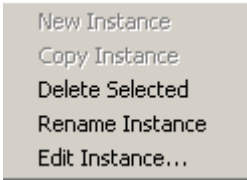
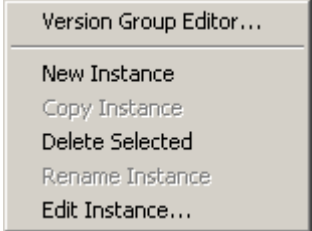
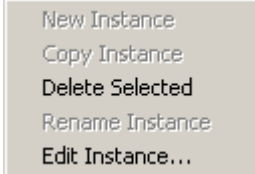
Class Name	Context Menu
Patch (PATCH)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul>

**Table 44: POLICY Instance Context Menus**

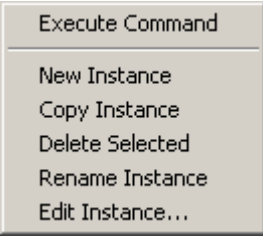
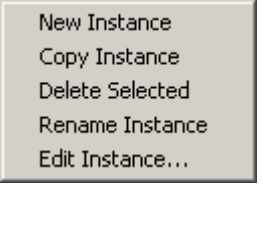
Class Name	Context Menu
All Class Instances within the POLICY Domain	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul>

**Table 45: SOFTWARE Instance Context Menus**

Class Name	Context Menu
Application (ZSERVICE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Notify Subscribers</li> </ul>
Application Packages (PACKAGE)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul> <hr/> <ul style="list-style-type: none"> <li>Copy Package...</li> <li>New Application Wizard...</li> </ul>
Auto Run (EXECUTE) Class Defaults (METACLAS) Dialog Services (DIALOG) HTTP Proxy (HTTP) Install Options (INSTALL) Scheduling (TIMER) SD Dependencies (SDDEP) SVR4 Dependencies (SVR4DEP)	<ul style="list-style-type: none"> <li>New Instance</li> <li>Copy Instance</li> <li>Delete Selected</li> <li>Rename Instance</li> <li>Edit Instance...</li> </ul>

Class Name	Context Menu
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)	
Registry Resources (REGISTRY)	
Version Groups (VGROUP)	
Versions (VERSION)	

**Table 46: SYSTEM Instance Context Menus**

Class Name	Context Menu
Application Manager (ZCOMMAND)	
Consoles (CONSOLE) Methods (ZMETHOD) Radia Processes (PROCESS) SQLTABLE (SQLTABLE) System Files (ZFILE)	



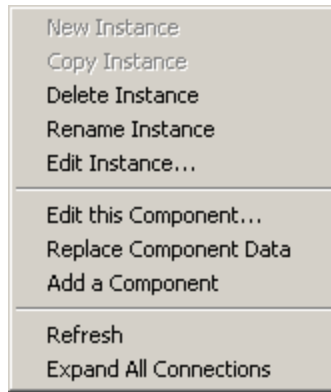
## C Modifying Component Instances (Non-Production Only)

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 System Explorer.
- Be able to add a component instance for a file or behavior to a software package, without having to republish the package.

▶ HP does not support this feature in production environments, and assumes no responsibility if a customer uses this feature where packages are actively being deployed to 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 the following figure.



**Figure 102: Instance shortcut menu showing Edit, Replace, and Add Component commands.**

## When to Use these Commands

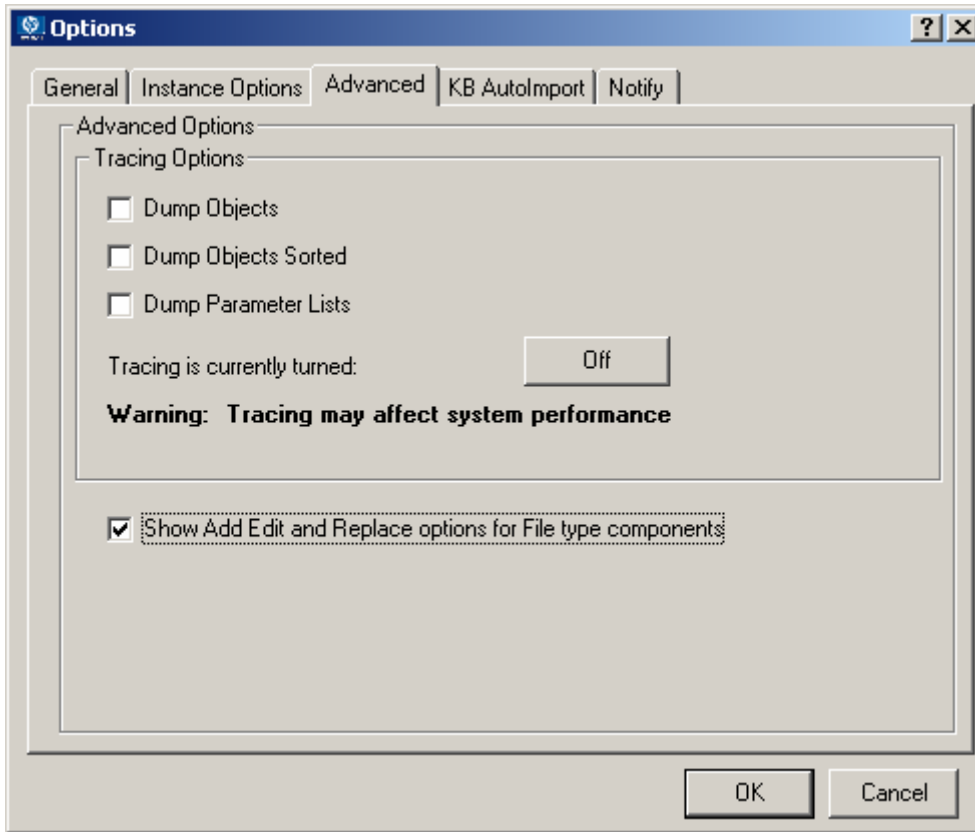
Examples of when you might want to use this option to modify component instances are listed in the following table.

**Table 47: When to Use the Advanced Edit Component Commands**

Command	Typical Use
Edit this Component	Change a serial number deployed with a file. Edit the contents of a <code>readme</code> file, document, spread sheet, logo image, or PDF.
Replace Component Data	Replace the entire contents of an existing file in a package with contents of another file (outside of Radia database).
Add a Component	Add a new file instance to a package, using another packaged file as a template.

## Accessing the Add, Edit, and Replace Component Commands

The following setting is available on the **Advanced** tab of the Options dialog box: **Show Add Edit and Replace Options for File type components**.

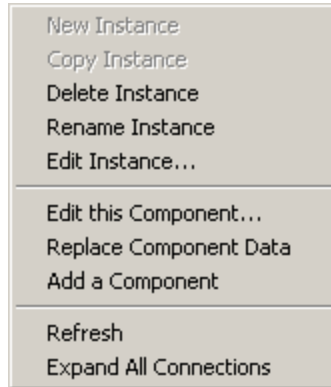


**Figure 103: Options window, Advanced tab includes Show Add Edit and Replace options for File type components.**

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.

- ▶ 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 menu for component instances.



**Figure 104: Instance shortcut menu showing Edit, Replace, and Add Component commands.**

## Editing a Component

The **Edit this Component** option permits you to select an editor of your choice with which to dynamically modify the distributed contents of a file instance. After you save the changes to the file, the modified file is sent (re-promoted) to the Configuration Server and the instance in the Radia database is updated.

- ▶ HP does not support this feature in production environments, and

assumes no responsibility if a customer uses this feature where packages are actively being deployed to 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 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 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 System Explorer.

Refer to the *Installation and Configuration Guide for the HP OpenView Application Manager Using Radia* for information on creating and promoting packages to the Radia database.

## To edit a component



The component editor does not know a component's target operating system. Prior to editing an instance, HP recommends that you note (or export) the values of the following attributes, as well as any other attributes, whose values should not change due to your edits:

ZRSCMEMM (PDS Member Name)  
ZRSCCRC (Resource CRC)  
ZRSCPADM (PDS AdminID)  
ZPUBVER (Packager/Publisher Version)

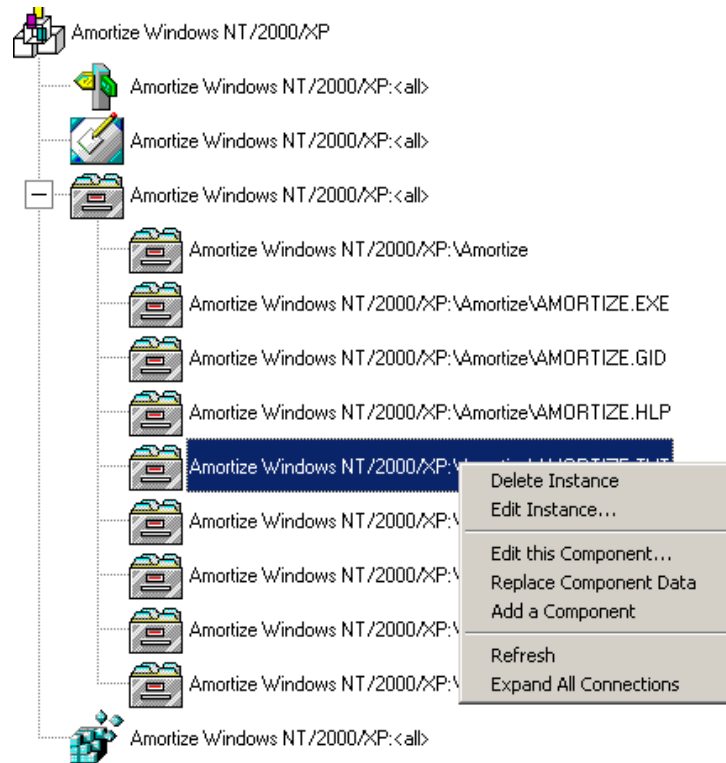
If the values of these attributes change as a result of the editing process, HP recommends that you restore the original values.

For example, if a component is targeted for a Windows operating system, the ZRSCMEMM, ZRSCCRC and ZRSCPADM attributes do not apply and should be null. After editing a component, if these attributes show values other than null, you should clear the inserted values. Likewise, the values for other attributes, such as ZPUBVER, should remain unchanged due to an edit. If changed, restore ZPUBVER to its original value.

- 1 In the 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.

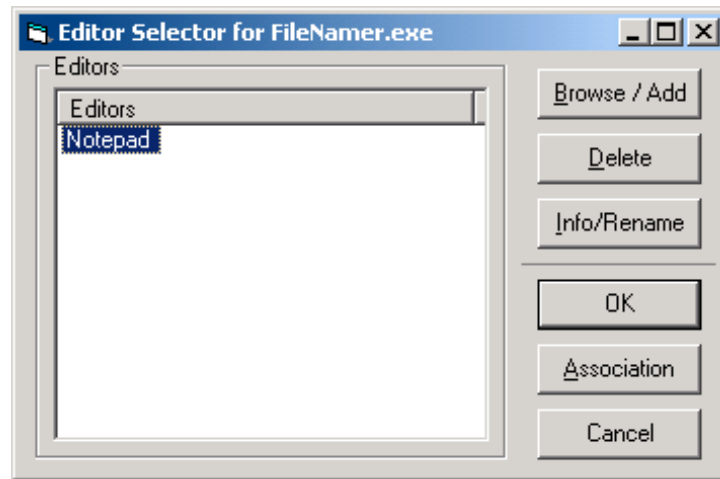
The following figure shows an example of the context menu for the `Amortize.TXT` readme file instance within the `Amortize NT/2000/XP` package. It includes the commands: **Edit this Component**, **Replace Component Data**, and **Add a Component**.



- 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`.



For details on using this dialog box, see Using the Editor Selector Dialog Box on page 329.

- 5 Double-click an editor from the **Editors** list box.

The selected editor is launched and opens the previously selected file component instance.

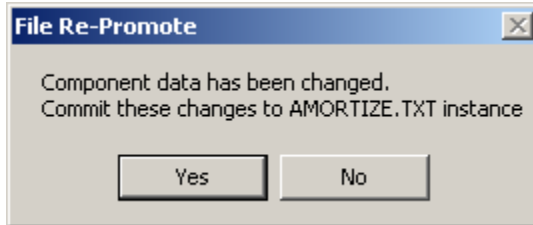
- 6 Make the required changes to the file's contents and save it with the *same name*.



If you save the file with a different name, the Radia component editor cannot make the changes to the file you selected from the System Explorer.

- 7 Close the editor.
- 8 If necessary, close the Editor Selector dialog box using **OK** to save the changes to the instance.

- 9 The File Re-Promote dialog box opens, and requests a confirmation that you want to commit the changes to the selected instance in the database.
  - Click **Yes** to 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.



After you commit the changes, the instance attributes for ZRSCDATE, ZRSCTIME, and ZRSCSIZE reflect the edited file size and date.

File Resources class Amortize \Windows NT/2000/XP:\Amortize\AMORTIZE.TXT Instance Attributes:		
Name	Attribute Description	Value
ZRSCNAME	Resource Name	&ZRSCCFIL
ZRSCCFIL	Resource File Name	\Amortize\AMORTIZE.T...
ZRSCMD	Mandatory/Optional on Client [M/O]	0
ZRSCVRFY	Verify Resource File on Connect	Y
ZRSCCONF	Confirm File Download [Y/N]	Y
ZRSCRASH	DOS File Attribute [R/A/S/H]	A
ZRSCSTYP	Server File Type [BINARY/TEXT]	BINARY
ZRSCMEM	PDS Member Name	21859BAE
ZRSCDATE	Resource Date Stamp - From Promote	20021023
ZRSCTIME	Resource Time Stamp - From Promote	16:27:40
ZRSCSIZE	Resource Size - From Promote	00000001193
ZRSCVERS	Resource Version - From Promote	
ZRSCCSTA	Client File Status	999
ZRSCCKPT	Resource Checkpoint	
ZRSCCRC	Resource CRC	

10/24/2002 12:23 PM





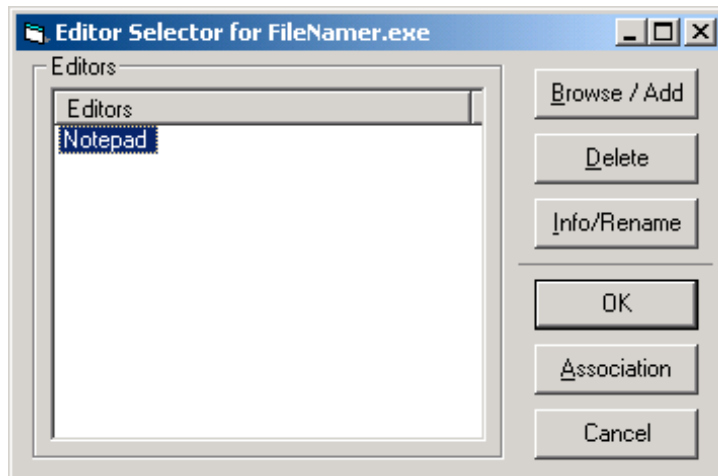
HP recommends that you compare all attributes' values against their original ones, and restore any values that changed unnecessarily. For example, you may need to clear values from OS-related attributes whose values were originally null. For details, see the cautionary text at the beginning of this procedure on page 324.

## 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 105: Editor Selector dialog box to Edit a Component.**

The following table summarizes the Editor Selector dialog box button actions.

**Table 48: Editor Selector Dialog Box Buttons**

<b>Button</b>	<b>Action</b>
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 <b>Editors</b> list below the highlighted entry.
Delete	Deletes a selected (highlighted) program from the <b>Editors</b> list.
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 <b>Editors</b> list.
OK	Opens the selected file (named in the title bar) with the currently highlighted editor.  Alternatively, you can double-click on the <b>Editors</b> 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 <b>Edit a Component</b> 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 System Explorer. Then, use the **Replace Component Data** command from within the 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.



HP does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are

actively being deployed to 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 System Explorer whose contents represent the replacement data for a packaged component file in the Radia database. The prepared file can have any name.

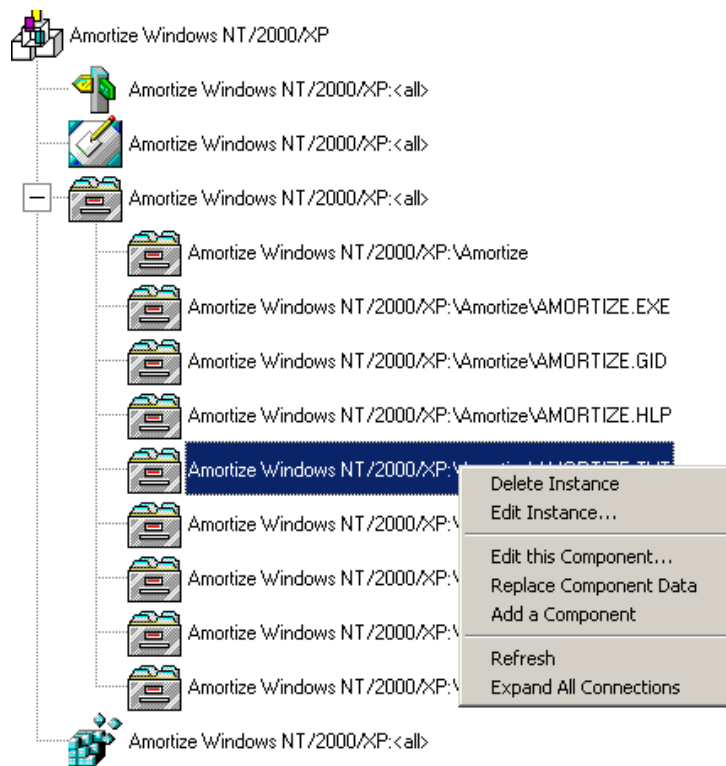


The entire contents of the prepared file will replace the entire data contents of the file selected from the System Explorer using the **Replace Component Data** command. Verify the contents of the files prior to continuing with this procedure.

- 2 In the 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**.

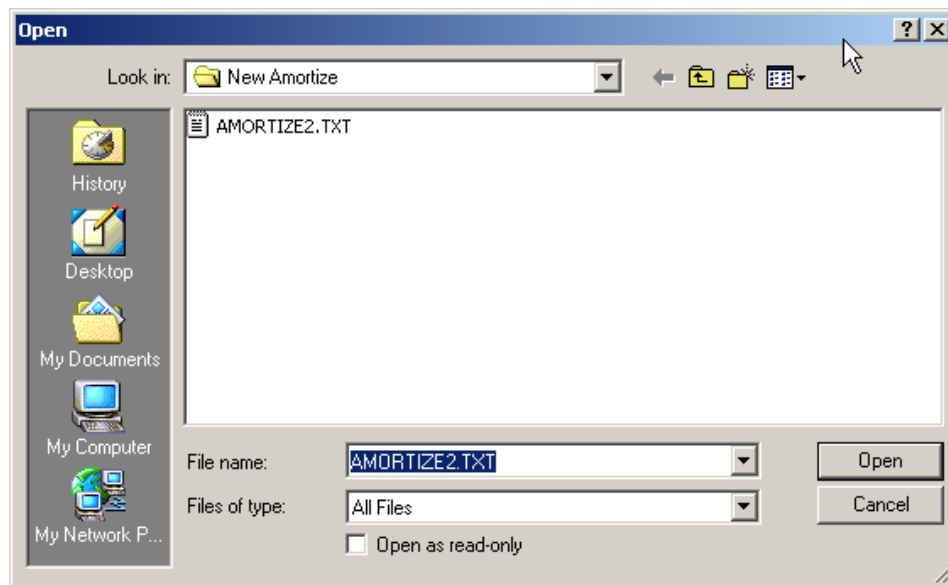


- 5 Select **Replace Component Data** from the shortcut menu.

A standard Windows Open dialog box opens.

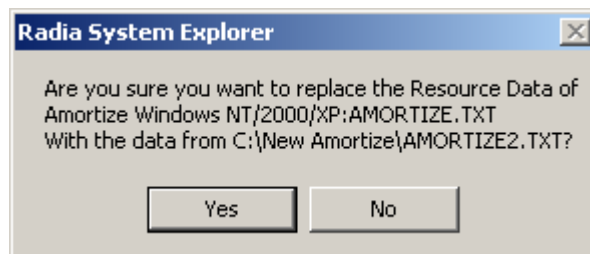
- 6 Navigate to the prepared file whose contents will replace the data for the file you selected from the System Explorer.

The following figure shows an example of navigating to a prepared file named `AMORTIZE2.TXT`.



- 7 Click **Open** to select the file with the replacement data.

The following dialog box opens to confirm the data replacement.



- 8 Click **Yes** to replace the data; click **No** to abort the data replacement.

If you click **Yes**, the 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.

If you click **No**, the replacement command is cancelled.

## Adding a Component

The **Add a Component** command is available for file and behavior components only. Files always belong to an owning package, so the most likely place to perform an Add is from the connection under a package. New files can be added by copying the properties of an existing published file; files cannot be created from scratch, that is, from files that have not been promoted from the Packager.



HP does not support this feature in production environments, and assumes no responsibility if this feature is used where packages are actively being deployed to 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.

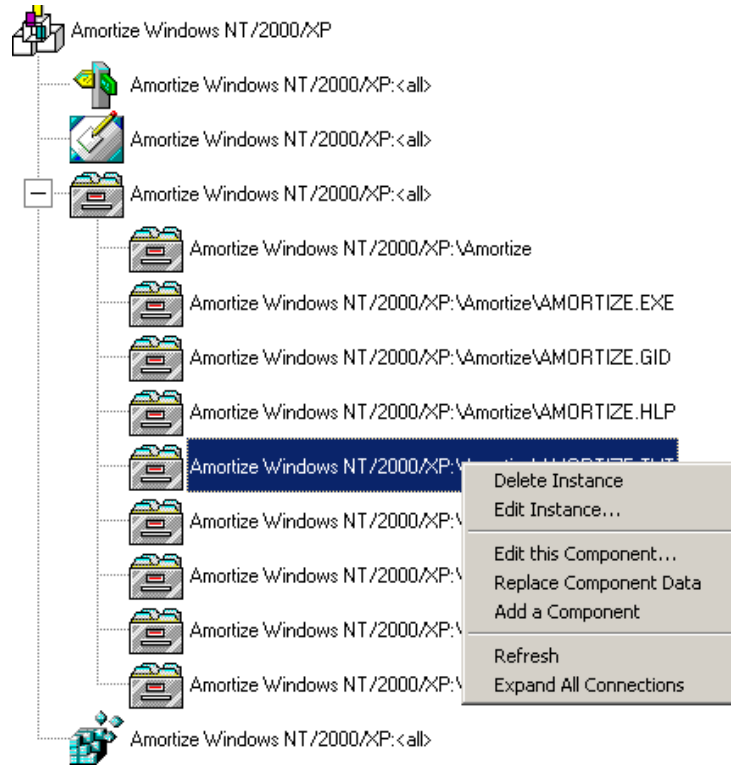
New files are added by selecting an instance to use as a template. The System Explorer will derive the path for the new instance from the template's LOCATION variable. Once a template is selected, right-click it to display the context menu and select **Add Component**. An Open File dialog box opens where 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 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**.

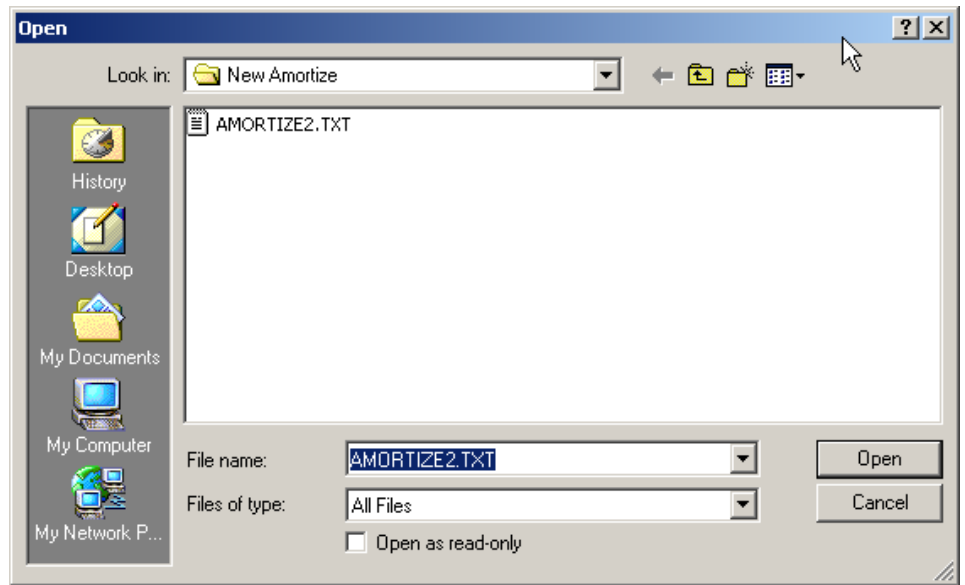


- 4 Select **Add a Component** from the shortcut menu.

A standard Windows Open dialog box opens.

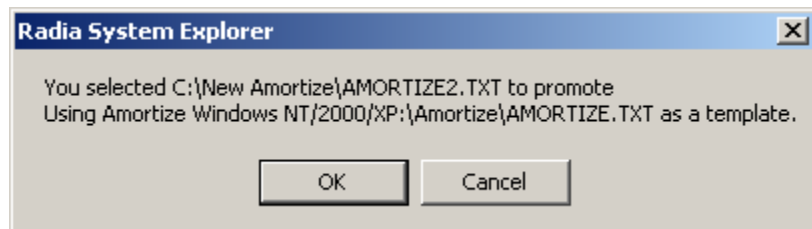
- 5 Navigate to the prepared file you want to add to the package.

The following figure shows an example of navigating to a prepared file named `AMORTIZE2.TXT`. In this case, we want to add this component to the package.



- 6 Click **Open** to select the file to be added with the same LOCATION as the template file.

The following dialog box opens to confirm the addition.



- 7 Click **OK** or **Cancel**.

If you click **OK**, the 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 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.

# Index

- `_BASE_INSTANCE_`
  - description, 65
  - inheritance, 65
- A Class is Copied check box, 42
- A New Class is Added check box, 42
- Add a Component
  - description, 334
  - when to use, 322
- Add/Remove Programs applet, 263, 264, 266
- ADMIN domain, 55
- Administrative Installation Point, 254, 268
- Advanced tab, 47
  - options described, 47
- alias name, 81
- always connection, 77
- application instance, creating, 136
- APPMGMT domain, 6, 57
- Association button, 330
- attribute
  - alias, 81
- Attribute Contains the Instance Alias Name, 81
- attribute type
  - connection, 77
  - expression, 76
  - method, 77
  - variable, 76
- AUDIT domain, 56
- BASELINE domain, 6, 57
- Browse/Add button, 330
- Build State File menu option, 217
- byte level differencing patching, 163
- Cancel button, 331
- CFM domain, 6, 57
- class
  - adding
    - attribute, 80
    - file, 334
    - instance, 88
  - attribute types
    - connection, 77
    - expression, 76
    - method, 77
    - variable, 76
  - connection types, 77
  - copying, 67
  - creating new, 70
  - deleting, 86
  - editing, 72
  - class connection, 77
  - Class Editor, 68, 71
  - class information, properties, 73
  - class type
    - component, 75
    - configuration, 75
    - none, 75
    - policy, 75
    - values, 75
  - CLIENT domain, 56
  - Client Explorer
    - accessing, 224
    - opening, 224
  - Client properties
    - execute, 78
    - protect, 78
  - Common Information Model
    - schema, 294
  - component
    - adding to a package, 335
    - editing, 325
  - component class instances, 208
  - component classes, 208
    - examples, 126

- filtering, 208
- component data, replacing, 331
- Configuration Server, version information, 303
- connection attribute type, 77
- connections
  - drag-and-drop for component class instances, 126
  - enabling, 239
- Contents tab of System Explorer help, 300
- context menus, 66, 305
- context-sensitive help, 298
- copying package, 269
- creating a state file, 216
- Current Feature Settings, 257
- customizing Radia, 25
- DATA1.MSI, 264
- Default Verify Options, 139
- defer notification, 196
- Delete button, 330
- Disable feature, 259
- distribution model
  - description, 23
  - elements
    - application files, 23
    - applications, 23
    - deployment destination, 23
    - deployment source, 23
    - users, 23
- domain
  - adding to a file, 66
- drag-and-drop connections
  - avoiding errors, 116
  - component class instances, 126
- drag-and-drop Notify, 189
- Edit Instance dialog box, 46
- Edit this Component
  - description, 324
  - when to use, 322
- editing a component, 325
- Editing Class dialog box, 42, 72
- Editing Instance dialog box, 90
- Editor Selector dialog box
  - buttons, 330
  - using, 329
- Editors list box, 327
- Email notification, 188
- event reporting, 143
  - keywords, 145
- Exist/Exist, 280
- exporting
  - application package to AKB, 217
  - state files, 214
- express filters, 210
  - customizing, 210
  - description, 210
- expression attribute type, 76
- Extended Verify Option, 140
- Feature Set Editor
  - Add/Remove Programs, 263, 264, 266
  - Current Settings, 257
  - dialog box, 250
  - Disable feature, 259
  - Features menu, 253
  - Hide feature, 259
  - navigating the tree, 250
  - shortcut menu, 249
- Features menu, 253
- file
  - adding, 334
  - adding a domain to, 66
- Filter Instances dialog box, 204, 208
- filter specifications
  - description, 202
  - examples, 202
  - valid expressions, 202
- filtering, 202
- filters
  - permanent
    - deleting, 206
    - description, 203
    - setting, 203
  - temporary
    - description, 206, 207
    - setting, 206, 207
- General tab, 41

- options described, 41
- Help menu, About Explorer, 303
- Hide and Disable shortcut menu, 259
- Hide feature, 259
- HP OpenView Using Radia
  - terminology, 24
- Ignore, 256
- IGNORE feature state, 256
- includes connection, 77
- Index tab of System Explorer help, 300
- Info/Rename button, 330
- inheritance, 65
  - \_BASE\_INSTANCE\_, 65
- Installed on First Use, 255, 257
- instance
  - adding, 88
  - adding from scratch, 89
  - copying, 88
  - deleting, 98
  - editing, 90
  - renaming, 97
- instance names, prefixes, 63
- Instance Options tab, 44
  - options described, 44
- instances
  - compound names, 63
  - prefixes, 63
  - using for instances, 63
  - prefixed names, 63
- Inventory Manager, 292
- KB AutoImport tab, 48
- LDAP directory, 293
- LICENSE file, 55
  - description, 55
- Lock the splitter bar distance to the right side of the screen, 43
- logon dialog box, 31
- MACHINE domain, 57
- Manager properties, table, 78
- mandatory applications, 141
- master service, 179
- master service, creating, 180
- MD5 signature, 161, 163
- menu-based Notify, 189
- message-digest algorithm, 161
- method attribute type, 77
- mouse
  - table of actions, 30
  - using, 30
- MSIEXEC, 271
- MSIFEATS, 249, 269, 271
- MSI-packaged products, 178
- MST, 271
- multiple instances
  - editing, 92
  - Editing Instance dialog box, 94
  - selecting, 93
- NAME attribute, 25
- Native Mode, 271
- New Application Wizard, 136
  - variables, 140
- Not Available, 255
- notify
  - by machine name, 50
  - specify STARTDIR, 50
  - specify UID, 50
- Notify
  - creating an audience, 189
  - description, 187
  - drag-and-drop, 189
  - file structure, 199
  - initiating, 188
  - menu-based selection, 189
  - remove an application, 193
  - retry, 201
  - Retry Manager, 201
  - send an e-mail, 192
  - Status window, 197
  - unsuccessful attempt, 201
  - update an application, 193
- NOTIFY file
  - description, 58
- Notify tab, 50
  - options described, 51

- NOVADIGM domain, 56, 57
- OfficeMaintenance Mode window, 266
- OK button, 330
- Optimize Service, 163
  - configuring, 165
  - using, 163
- optional applications, 141
- OS domain, 57
- owning package, 208
- package, duplicating, 269
- patch building, 173
- PATCH class, 161, 176
- PATCH domain, 56, 161, 176
- Patch Manager, 292
- patch-building restrictions, 173
- PATCHMGR domain, 57
- permanent filters, 203
  - deleting, 206
  - description, 203
  - setting, 203
- POLICY domain, 56
- Populate Windows Installer Methods check box, 139
- pop-up menus, how to open, 305
- PRDMAINT domain, 56
- prefixes, using in compound names, 63
- PRIMARY file
  - ADMIN domain, 55
  - APPMGMT domain, 6, 57
  - AUDIT domain, 56
  - BASELINE domain, 6, 57
  - CFM domain, 6, 57
  - CLIENT domain, 56
  - domains of, 59
  - MACHINE domain, 57
  - NOVADIGM, 57
  - NOVADIGM domain, 56
  - OS domain, 57
  - PATCH domain, 56
  - PATCHMGR domain, 57
  - POLICY domain, 56
  - PRDMAINT domain, 56
  - SOFTWARE domain, 56
  - SYSTEM domain, 56
  - USAGE domain, 57
- PROFILE file
  - description, 58
- Prune Below, 61
- Radia
  - customizing, 25
  - mouse actions, 30
- Radia Application Manager, 292
- Radia client, version and build information, 303
- Radia database
  - adding
    - class, 67
      - copying, 67
    - class attribute, 80
    - domain, 66
    - file, 334
    - instance, 88
      - copying, 88
      - from scratch, 89
  - attribute, 25
    - definition, 25
    - example, 25
  - attribute value, 25
    - definition, 25
    - example, 25
  - changing tree view, 59
  - class, 24
    - definition, 24
    - example, 24
  - class instance, 24
    - definition, 25
    - example, 25
  - contents of, 54
  - deleting a class, 86
  - deleting an instance, 98
  - domain, 24
    - definition, 24
    - example, 24
  - drop down lists, 84
  - editing, 66, 305
    - class, 72
    - instance, 90

- file, 24
  - definition, 24
  - example, 24
- filtering, 202
- flag attribute, 82
- flag set attribute, 83
- hierarchy, 32
- Manager Properties table, 78
- navigating the tree structure, 59
- navigation history, 61
  - discarding, 61
  - Prune Below, 61
- new class
  - creating, 70
- optional editing, 82
- renaming an instance, 97
- structure, 24
- tree view, 32
- Version Groups
  - adding a Version instance, 153
  - assigning a Version instance, 156
  - deleting a Version instance, 159
  - editing, 159
  - removing a Version assignment, 159
- Radia MSI methods, 140
- Radia Registry File Editor. See Registry File Editor
- Radia Service Groups, 178
- Radia SQL database, 290, 292, 293
- Radreg32 Delete operation
  - setting options, 281
- radskman, 179
- Registry File Editor, 276
  - accessing, 276
  - active option, 280
  - context menu options, 280
  - delete always option, 281
  - delete default option, 281
  - delete never option, 281
  - deploy/verify option, 280
  - exiting the dialog box, 285
  - inactive options, 280
  - navigating the tree, 279
  - new key option, 281
  - new value option, 281
  - setting properties, 280
- registry key, editing, 282
- Registry Resource instance, 276
- Rename Instance dialog box, 97
- Replace Component Data
  - description, 331
  - when to use, 322
- Reporting Server, 292
  - overview, 292
- requires connection, 77
- resolution process
  - description, 224
  - evaluating stop expressions, 232
- RIMSFSCAN, 294
- RIMWBEM, 294
- Run all from My Computer, 254
- Run all from Source, 254
- Run from My Computer, 253
- Run from Source, 254
- Search tab of System Explorer help, 300
- security dialog box, 31
- Server Management Using Radia
  - optional domains in PRIMARY file, 6, 57
- Service Groups
  - master service, 178
- Service Groups, creating, 179
- Service Optimization, 160
- SHOWINCP variable, 264, 265
- SOFTWARE domain, 56, 60
- specifying a domain name, 195
- state file
  - creating, 216
  - exporting, 214
- stop expressions, 232
- SYSTEM domain, 56
- System Explorer
  - capabilities, 22
  - context menus, 66, 305
  - control features, 40
  - database, changing tree view, 59

- filtering the database, 202
- navigating the tree structure, 59
- opening, 31
- options, 40
  - Advanced tab, 47
  - General tab, 41
  - Instance Options tab, 44
  - KB AutoImport tab, 48
  - Notify tab, 50
- pop-up menus, 305
- window appearance, 33
  - re-sizing, 33
- System Explorer help
  - Contents tab, 300
  - Index tab, 300
  - Search tab, 300
- System Explorer toolbar buttons, 39
  - described, 39
- System Explorer, build state file, 214
- temporary filters
  - deleting, 206
  - description, 206, 207
  - setting, 206, 207
- terminology in HP OpenView Using, 24
- tool tips, description, 32
- transform file, 271
- transient classes, 74
- tree view, manipulating, 59
- Unconditional/Enforced, 280
- Unconditional/Exist, 280
- Update Features window, 267
- USAGE domain, 57
- Usage Manager, 292
- USER class, 56
- USERID attribute, 25
- variable attribute type, 76
- varsets subdirectory, 216
- Verify Options, 139
- Version Group
  - adding a Version instance, 153
  - assigning a Version instance, 156
  - deleting a Version instance, 159
  - editing, 159
  - removing a Version assignment, 159
- Version Group Editor, description, 149
- WbEM namespace, 293
- ZCONFIG, 216
- ZHDWMEM, 216
- ZOBJPTCH variable, 161
- ZPUBVER
  - restoring original value, 325
- ZRSCCRC
  - restoring original value, 325
- ZRSCMMEM
  - restoring original value, 325
- ZRSCPADM
  - restoring original value, 325
- ZSERVICE instance, 56