HP OpenView Application Manager Using Radia

Radia Application Manager Guide

Software Version: 4.0

for the Windows operating system

Legal Notices

Warranty

Restricted Rights Legend

Copyright Notices

Trademark Notices

Acknowledgements

Technical Support

Please select Support & Services from the following web site:

<http://www.hp.com/managementsoftware/services>

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

The support site includes:

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

About this Guide

Who this Guide is for

This guide should be used by Radia systems administrators who want to install and use the Radia Application Manager for Windows.

What this Guide is about

This guide contains a comprehensive description of the Radia Application Manager, version **4.0**. It includes in depth discussions of the following topics:

■ How to install the Radia Client and Radia Administrator Workstation.

■ How to publish applications.

■ How to use Radia Client Operations Profiles.

■ How to implement entitlement policies.

■ How to prepare services.

■ How to deploy applications.

■ What Radia Client objects are.

> **Note**
>
> To use the new features of the Radia Application Manager, Version 4.0, you need the Radia Database, Version 4.0 or above.

Summary of New Features

This printing of the *Radia Application Manager for Windows* version 4.0 contains the following new features for the following chapters.

Chapter 2:
Radia Client Installation

4.0 Page 36, *[PROPERTIES] Section of INSTALL.INI*: The following have been added to the properties section of the install.ini file: NVDSTARTWMICFGMGR, NVDRAMSHORTCUT, NVDRAMSTARTMENUSHORTCUT; NVDRAMCONNECT, NVDMAINTDIR, NVDNOTIFYINTERACT, NVDREDIRECTORINTERACT, NVDSCHEDULERINTERACT.

4.0 Page 46, *Specifying the Radia Client Features to Install*: The Radia Patch Manager and Radia OS Manager clients have been added to the Radia Client installation. Only install the clients for which you are licensed.

Chapter 4:
Packaging Applications and Content

4.0 Page 116: The Radia Publisher for Radia Version 3.x has been renamed the Radia Packager for Radia Version 4.x.

Chapter 6:
Configuring Client Operations Profiles

4.0 Page 261: *Configuring Client Operations Profiles* is a new feature of version 4.0 for the Radia Application Manager client. Client Operations Profiles allow you to create redundancy and fail over capabilities for Radia Configuration Servers and Radia Proxy Servers, control hardware scans, diagnostic settings, and user interface options.

Chapter 7:
Preparing Services

4.0 Page 306, *Enhanced Machine User Services for Radia Software Manager*: Radia 4.0 has augmented its machine/user services.

Chapter 8:
Deploying Services

4.0 Page 349, *Table 8.3 ~ Radskman Machine/User Parameters*: The following parameters have been added to radskman: *catexp* to filter applications, *machfreq* for thin clients, and *mnt* to control when Radia Self Maintenance is applied.

4.0 Page 350, *Table 8.4 ~ Radskman Client Operations Profiles Parameters*: The following parameters have been added to radskman for use with Client Operations Profiles: *cop*, *datauri*, *product*, and *rcsuri*.

4.0 Page 352, *Table 8.5 ~ Radskman Process Parameters*: added the upd parameter, which prevents updates to applications during the Client Connect session.

4.0 Page 358, *Table 8.6 ~ Scheduling (TIMER) Attributes*: *NETAVAIL, RETRYRC, RETRYFLG, RETRYINT, RETRYLMT* attributes have been added to the TIMER class to allow for retry if a timer event fails.

4.0 Page 362, *Table 8.7 ~ Syntax of ZSCHDEF Attribute*: *MONTHLY, MONTHDAY*, and *STARTUP* values have been added to the ZSCHDEF attribute in the Scheduler (TIMER) class.

4.0 Page 399, *Radia Self Maintenance:* is now supplied to the customer in the form of export decks. The Radia 4.0 clients use the PRDMAINT domain.

4.0 Page 402, *Radia System Tray*: Radia Application Manager now supports French and German. If the operating system is set to French or German, Radia System Tray will display in the specified language.

4.0 Page 412, *Local AIP support for the Radia MSI Redirector:* allows you to add another level of redirection at the local file system level. When using this method, MSI is told that the AIP is local, and requests files directly from the file system.

Chapter 9:
Radia Client Directories and Objects

4.0 Page 422, *Table 9.2 ~ Core Client Objects*: added a row for SMINFO.

4.0 Page 429, *Radskman Execution (PREFACE):* The following client objects are documented in this guide: ZCONFIG, SAPSTATS, SYNOPSIS, PREFACE, and SMINFO. *Radskman* Execution (PREFACE) was introduced in version 3.1. *SAPSTATS* and *SYNOPSIS* are used with Client Operations Profiles. *Systems* Management Information (SMINFO) is a new object that includes unique computer information taken from the client computer's BIOS tables.

4.0 Page 430, *Systems Management Information (SMINFO):* new section.

Appendix B:
Radia Thin Client

4.0 Page 443, *Appendix B: Radia Thin Client Support*: Radia supports thin client computers.

Editorial Improvements

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

Conventions

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

Table P.1 ~ Styles		
Element	**Style**	**Example**
References	*Italic*	See the *Publishing Applications and Content* chapter in this book.
Dialog boxes and windows	**Bold**	The **Radia System Explorer Security Information** dialog box opens.
Code	`Andale Mono`	`radia_am.exe`
Selections	**Bold**	Open the **\Admin** directory on the installation CD-ROM.

Table P.2 ~ Usage		
Element	**Style**	**Example**
Drives (system, mapped, CD)	Italicized placeholder	*SystemDrive*:\Program Files\Novadigm might refer to C:\Program Files\Novadigm on your computer. *CDDrive*:\client\radia_am.exe might refer to D:\client\radia_am.exe on your computer.
Files (in the Radia Database)	All uppercase	PRIMARY
Domains (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE May also be referred to as the SOFTWARE domain in the PRIMARY file.
Classes (in the Radia Database)	All uppercase	PRIMARY.SOFTWARE.ZSERVICE May also be referred to as the ZSERVICE class in the SOFTWARE domain in the PRIMARY file.

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

Table P.3 ~ Terminology*	
	* Depends on the context. May not always be able to substitute.
Term	**May also be called**
Application	software, service
Client	Radia Application Manager and/or Radia Software Manager
Computer	workstation, server
NOVADIGM domain	PRDMAINT domain **Note**: As of the 4.0 release of the database, the NOVADIGM domain is being renamed the PRDMAINT domain. Therefore, if you are using an earlier version, you will see the NOVADIGM domain in the database.
Radia Configuration Server	Manager, Active Component Server
Radia Database	Radia Configuration Server Database

Contents

7 Preparing Services.. 299

1

Introduction

At the end of this chapter, you will:

■   Understand the components of Radia.

■   Be familiar with the structure of the Radia Database.

■   Understand suggested deployment strategies.

Note
To use the new features of the Radia Application Manager, Version 4.0, you need the Radia Database, Version 3.12 or above.

About Radia Technology

Radia technology provides high levels of adaptability, flexibility, and automation. *Adaptability* comes from the embedded intelligence of platform-independent object-oriented technology. *Flexibility* is provided by the media-independence of Radia technology that enables content to be easily revised and customized. And its solutions *automate* digital asset management across virtually any kind of network. The following bullets describe each of these distinctive capabilities that are essential to Radia technology:

- **The Embedded Intelligence of Object-Oriented Technology.**
  Object-oriented technology transforms software and content from file-based media into self-aware, platform-independent, intelligent objects that automatically assess the environment into which they are deployed, and personalize, install, update, and repair themselves accordingly. In other words, as intelligent objects, they know what they need for a particular device or user, where to get it, when they need to change, how to change themselves, and how to repair themselves.

- **Revisable Packaging for Revisable Content.**
  Radia technology enables revision and customization of software and content at any midstream point in the publisher-to-subscriber deployment process. Because Radia technology transforms software and content into objects, these objects can be easily modified midstream – subtracted from, added to, or reconfigured – simply by packaging them with other objects or new configuration information. With revisable packaging, value-added service providers and IT administrators may customize standard published software offerings for the needs of their particular users without having to unpack and repackage everything.

- **Self-Managing Infrastructure.**
  The object-oriented intelligence of Radia technology incorporates a self-managing infrastructure. This capability begins with network-independence, with Radia technology flexibly supporting any deployment environment, whether client/server, local, wide, or virtual area network, intranet, extranet, or the Internet. Furthermore, we supports whatever distribution media make sense for the target audience and the provider (which might be a software publisher, application service provider (ASP), Internet service provider (ISP), provider of enterprise application integration (EAI) services, e-business integrator, e-commerce component provider, or in-house IT administrator).

In the Internet age in which software is fundamental to the ability of businesses to compete, change is a constant state, and audience diversity has grown beyond the capacity of older technologies to manage. Our technology provides the necessary automation, adaptability, and flexibility to solve the software management challenge.

Distribution Models

Radia manages the distribution of digital assets based on your *distribution model*. A distribution model records the identities and intended configurations of the desktop computers whose configurations are managed by Radia. The distribution model can be simple or complex, and may also be referred to as the desired state.

At a minimum, a Radia distribution model includes the following five elements:

■ **Users**
The identity of the computers being managed. This can be either a computer name or a user name.

> **Note**
>
> The term *computer* is used to refer to a workstation or server.

■ **Applications**
The software that is being managed.

■ **Application Resources**
The components that make up the applications.

■ **Deployment Source**
The location where the application components are stored, such as on a Radia Proxy Server or Radia Configuration Server, so they can be distributed to the users.

■ **Deployment Destinations**
The location to which the application and its files will be distributed such as desktop computers, servers, PDAs, and laptops.
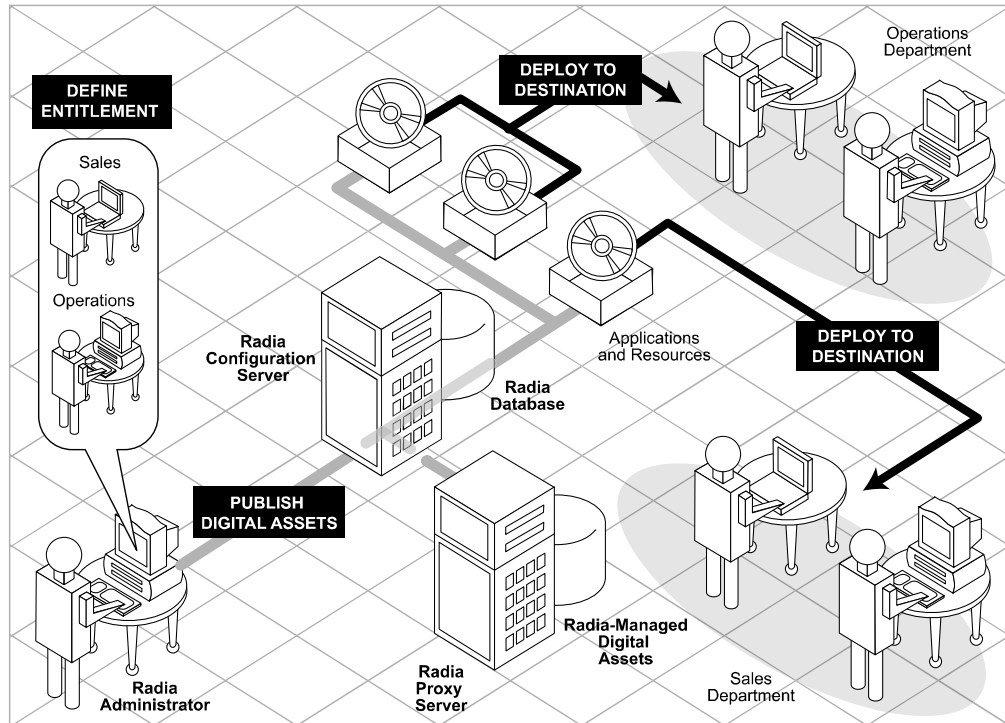
Figure 1.1 ~ Elements in a distribution model.

Use Radia to manage all of these components. You will publish *packages* of digital assets, determine entitlement policy, and define how the packages will be deployed.

> **Note**
>
> A package is a unit of distributable software or data.

The Radia Database

The Radia Database, located on the Radia Configuration Server, stores the information needed to create the 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 policies determining which subscribers are assigned to which packages.

■ Security and access rules for Radia administrators.

Use the Radia System Explorer to view and manipulate the Radia Database. The Radia Database is hierarchically structured, and its components consist of files, domains, classes, instances, and attributes.

Table 1.1 ~ Radia Database Structure	
Term	**Description**
File	Highest level in the hierarchy of the Radia Database. Groups like domains together. **Example**: The PRIMARY file is used to define and maintain the distribution model. This is one of the pre-configured files distributed with Radia.
Domain	Logically partitions a Radia file. Groups like classes together. **Example**: The POLICY domain contains the classes needed to create users and groups.
Class	A category of the distribution model. The class is a template for the attributes needed to create an instance of the class. Refer to the *Radia Database Reference Manual* for information on the structure and usage of Radia classes. **Example**: The USER class of the POLICY domain defines subscribers of Radia-managed applications. It defines all of the attributes necessary to identify the client computer to be managed by Radia.
Class Instance or Instance	An object containing a specific occurrence of a class. This is analogous to a row in a relational data table, or a record in a traditional flat file. The attributes of a class instance object contain data describing one specific entity of that class. **Example**: A USER instance is an object created from the USER class, containing the information needed to identify a subscriber's client computer.
Attribute Attribute Value	An attribute is a data element 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**: The NAME attribute of a USER class contains the name of the subscriber, and the USERID attribute contains the User ID, as specified by the Radia administrator.

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

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

■ The PRIMARY file is where you will find most information regarding software management. Within the PRIMARY file, there are seven default domains.

- Use the ADMIN domain to define administrative rights and rules for connecting classes.

- Use the AUDIT domain to configure tasks that will inventory, or audit, client computers' assets. See the *Radia Inventory Manager Guide* for more information.

- Use the CLIENT domain to configure Client Operations Profiles. This includes defining which Radia Configuration Servers, Radia Proxy Servers, and Radia Staging Servers the client computer can you. For more information, see the *Configuring Client Operations Profiles* chapter.

- Use the PRDMAINT domain to store packages for self-maintenance that are supplied by HP. This domain should only be used for the deployment of Radia Client maintenance packages. See *Radia Self Maintenance* in the *Deploying Services* chapter in this book for more information.

  > **Note**
  >
  > In previous versions of the product, the maintenance functions were in the NOVADIGM domain.

- Use the PATCH domain to store information for binary patching of files associated with Service Optimization. See the *Radia System Explorer Guide.*

  > **Note**
  >
  > The Radia Patch Manager uses a different domain called PATCHMGR for managing security patches.

- Use the POLICY domain to create users and groups, and to assign users to groups. See the *Implementing Entitlement Policy* chapter in this book for more information.

- The SOFTWARE domain contains information about the software being managed and the methods used to deploy the software. See *Preparing Services* and *Deploying Services* in this book.

- The SYSTEM domain contains administrative and process control definitions.

■ As you begin to use Radia, the PROFILE file appears. This file contains information collected from client computers. The file appears after the first client computer has registered with the Radia Configuration Server. This information is used to connect to computers to deploy software managed by Radia, and to see the configuration of the client computer. The PROFILE file is discussed in the *Radia Client Directories and Objects* chapter in this book.

■ The NOTIFY file contains information about attempts by the Notify function to update, remove or e-mail subscribers. This file appears after the first attempted Notify. For more information about Notify, see the *Deploying Services* chapter in this book.

Radia Infrastructure

Use Infrastructure components to take full advantage of the ability to manage your enterprise's computing environment. Depending on your configuration, your infrastructure may be enhanced by any combination of these components. The Radia components can be divided into four categories.

- Radia Management Applications
- Radia Management Infrastructure
- Radia Extended Infrastructure
- Radia Management Extensions

Some of the basic Radia Infrastructure components are described below. For more information on all of the Radia products, see the *Radia Getting Started Guide* or the HP OpenView web site.

Radia Configuration Server

The Radia Configuration Server is part of the Radia Management infrastructure, and resides on a single server or across a network of servers. Applications and information about the subscribers and client computers are stored in the Radia Database. The Radia Configuration Server distributes packages based on policies established by the Radia administrator. See the *Radia Configuration Server Guide* (also known as the *Manager Guide*) for more information.

Radia Management Portal

The Radia Management Portal is a web-based interface that you can use to manage your infrastructure. The Radia Management Portal is part of the Extended Infrastructure. Whether you are already using Radia, or are just beginning, you can use the portal to create a graphical representation of your infrastructure. See the *Radia Management Portal Guide* for more information.

Radia Proxy Server

If you want to reduce the load on the Radia Configuration Server, or store your digital assets closer to your client computers, consider using a Radia Proxy Server. The Radia Proxy Server stores a copy of the digital assets that are available to subscribers attached to the Radia Proxy Server. The Radia Proxy Server is also part of the Radia Extended Infrastructure. Evaluate the potential benefits for each server and its attached subscribers individually. For more information, refer to the *Radia Proxy Server Guide*.

> **Note**
>
> Contact your HP representative for details on the Radia Management Portal and the Radia Proxy Server.

Radia Administrator Workstation

Radia comes with a set of tools used to carry out software management functions. You should become very familiar with these tools. This is part of the Radia Management Infrastructure. These include:

■ **Radia Packager**
Use the Radia Packager to create groups of components, called *packages,* and promote them to the Radia Configuration Server. See *Packaging Applications and Content* in this guide for more information.

■ **Radia System Explorer**
Use the Radia System Explorer to view and to manipulate the Radia Database. In addition to this publication, see the *Radia System Explorer Guide* for more information.

■ **Radia Client Explorer**
Use the Radia Client Explorer to view and to manipulate Radia objects on the client computer.

■ **Radia Screen Painter**
Use the Radia Screen Painter to create custom dialog boxes.

■ **Radia Publisher**
Use the Radia Publisher to publish Windows Installer files. See the *Radia Publisher Guide* for more information.

Management Applications

Management Applications (clients) allow you to automate deployment, update, repair, and deletion activities, and inspect hardware and software. Install the Radia Management Applications onto the subscriber's computer.

There are three types of Management Applications available for communicating with the Radia Configuration Server. Install the only those clients for which you have obtained a license. The client software is located on the Management Applications CD-ROM.

■ **Radia Application Manager**
Use this client to distribution mandatory applications throughout the enterprise. This client is described in this book.

■ **Radia Software Manager**
Subscribers install, remove, or update optional applications that are available to them in a service list. For more information, see the *Radia Software Manager Guide.*

■ **Radia Inventory Manager**
This client allows you to collect hardware information and send it to the Radia Inventory Manager for collection and reporting. See the *Radia Inventory Manager Guide* for details.

■ **Radia Patch Manager**
The Radia Patch Manager analyzes and manages security patches. See the *Radia Patch Manager Guide*.

■ **Radia OS Manager**
The Radia OS Manager controls the provisioning of operating systems. See the *Radia OS Manager Guide*.

If you install both the Radia Software Manager and Radia Application Manager feature sets, you decide if an application is mandatory or optional, and specify who controls the installation of the application. By adding the Radia Inventory Manager, you can also find out the hardware and software configurations of the client computer.

About This Guide

This guide covers the *suggested* implementation for the Radia Application Manager. Although you will tailor this strategy to meet your organization's needs, we recommend that you review this guide for a comprehensive understanding of the Radia Application Manager. At the start of each chapter, you will find the following diagram to help you locate where you are in the implementation. The appropriate area will be shaded. Before you can manage software, you must install the Radia Client and the Radia Administrator Workstation.
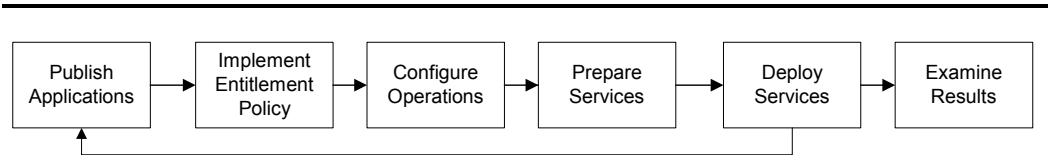
Figure 1.2 ~ Tasks completed in this guide.

The *Radia Application Manager Guide* covers the following:

■ **Installing the Radia Clients**
This chapter describes the how to install the Radia Clients.

■ **Installing the Radia Administrator Workstation**
This chapter describes how to install the Radia Administrator Workstation.

■ **Publishing Applications and Content**
This chapter describes how to publish applications using Component Selection Mode and Installation Monitor Mode.

■ **Implementing Entitlement Policy**
This chapter shows you how to define users and groups, and how to connect them to the appropriate applications.

■ **Configuring Client Operations Profiles**
This chapter explains how to configure your clients to use the most appropriate Radia Configuration Servers and Radia Proxy Servers, provide for fail over capabilities, and configure your Radia Client.

■ **Preparing Services**
This chapter describes services options such as restarting the client computer and implementing applications that have machine and user specific components.

■ **Deploying Services**
This chapter explains how to deploy applications to your client computers.

■ **Radia Client Objects and Directories**
This chapter shows you where to find and how to examine the results of your Radia implementation.

Summary

- Radia gives you the flexibility and control to efficiently manage desktop software.
- The Radia Database includes all the information needed to manage your software.
- We provide suggested deployment strategies that you should tailor to your organization's needs.

2

Installing the Radia Clients

At the end of this chapter, you will:

■ Understand the system requirements for installing Radia Clients.

■ Know how to customize the installation process.

■ Be able to modify the [PROPERTIES] section of the INSTALL.INI in order to customize the behavior of the installation process.

■ Be able to modify the [ARGS] section of the INSTALL.INI in order to customize the behavior of the Radia Software Manager.

■ Be able to modify the [OBJECTS] section of the INSTALL.INI in order to specify Radia objects to be created on the client computer.

■ Know how to use the Installation Wizard.

■ Know how to remove and repair Radia Clients using the Installation Wizard and command lines.

■ Know how to modify the installation of the Radia Clients using the Installation Wizard and command lines.

■ Know how to use pre-install script to customize MSI properties that affect installation.

■ Know how to use a post-install script to run processes after installing Radia Clients.

The Radia Client installation program uses Microsoft Windows Installer. The program consists of one MSI package with three feature sets—Radia Application Manager, Radia Software Manager, and Radia Inventory Manager.

Caution
Install only the Radia Clients for which you have licenses. If you do not have a license, the Radia Client will not authenticate with the Radia Configuration Server.

System Requirements

■   Windows 95, 98, NT 4 with Service Pack 6a, 2000, Server 2003, or XP.

Note
The Radia Software Manager 4.x client is not available for computers running Windows 95, as it requires Microsoft .NET runtime.

■   Microsoft .NET runtime *(required for Radia Software Manager only)*.
The .NET installation program is available in the **\DotNet** folder on the Radia Client CD-ROM. If .NET does not exist on the client computer, the .NET installation program runs automatically. Microsoft .NET requires Microsoft Internet Explorer 5.01 or later.

■   MS Windows Installer Version 2.0 or higher.
The MSI 2.0 installation program is available in the **\MSI** folder on the Radia Client CD-ROM. If Windows Installer does not exist, or if an earlier version is detected on the client computer, the MSI 2.0 installation program runs automatically.

■   TCP/IP connection to a computer running the Radia Configuration Server.

■   WMI (*required for the Radia Inventory Manager only)*.
The WMI installation program is available in the **\WMI** folder on the Radia Client CD-ROM. If WMI does not exist on the client computer, the WMI installation program runs automatically.

■   For Windows NT, 2000, Server 2003, or XP, you must have administrator rights to the computer to install the Radia Clients.

About the Installation Process

Whether the Radia Client installation program is distributed as an executable (setup.exe) or an Administrative Installation Point (AIP), the installation process is the same. You can customize many aspects of the installation including which Radia Clients to install and to what directory the installation files should be copied. If you want to customize the installation process, you should be familiar with the following files.

■   **setup.exe**
     setup.exe is stored in the **\win32** directory on the Radia Client CD-ROM. It accepts any standard Windows Installer command line parameters and passes them to the Windows Installer service.

     You can also create an AIP for network installations.

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

     To create the Windows Installer AIP in a specified target directory, type:

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

     The target directory contains RADIA40.MSI, the installation folders, setup.exe, and any files (such as INSTALL.INI or Visual Basic scripts) stored in the same directory as setup.exe. Next, copy the **\DotNet** and **\MSI** folders into the target directory.

> **Note**
>
> If you do not have Windows Installer version 2.0 installed, setup.exe looks for the standard MSI installation programs (INSTMSI.EXE or INSTMSIW.EXE) in the current directory and in the MSI subdirectory. If found, the appropriate MSI installation will run before the Radia Client installation.

■   **RADIA40.MSI**
     RADIA40.MSI is the MSI database file, which contains the default configuration information for the installation. This file is stored in the **\win32** directory on the Radia Client CD-ROM.

■   **INSTALL.INI**
     Use INSTALL.INI to customize the installation or the Radia Client arguments file, or to create or set attributes for Radia objects. Settings in INSTALL.INI override the defaults stored in RADIA40.MSI.

     A sample INSTALL.INI is available in the **\win32\samples** directory on the Radia Client CD-ROM.

■ **args.xml**
args.xml is the Radia Software Manager arguments file created from information stored in the [ARGS] section of INSTALL.INI. This file, stored in IDMLIB on the client computer, controls the behavior of the Radia Software Manager. The default directory for IDMLIB is *SystemDrive*:\Program Files\Novadigm\Lib\.

■ **Pre-install scripts**
(*Recommended for experienced users only*.)
Use custom Visual Basic scripts to customize MSI properties that affect the installation. For an example of a simple script, see *Using a Pre-Install Script* on page 79. This script is available in the **\win32\samples** directory on the Radia Client CD-ROM.

■ **Post-install scripts**
(*Recommended for experienced users only*.)
Use custom Visual Basic, REXX, or Tcl scripts to run processes such as the first Client Connect. For an example, see *Using a Post-Install Script* on page 81. The complete version of this script is available in the **\win32\samples** directory on the Radia Client CD-ROM.
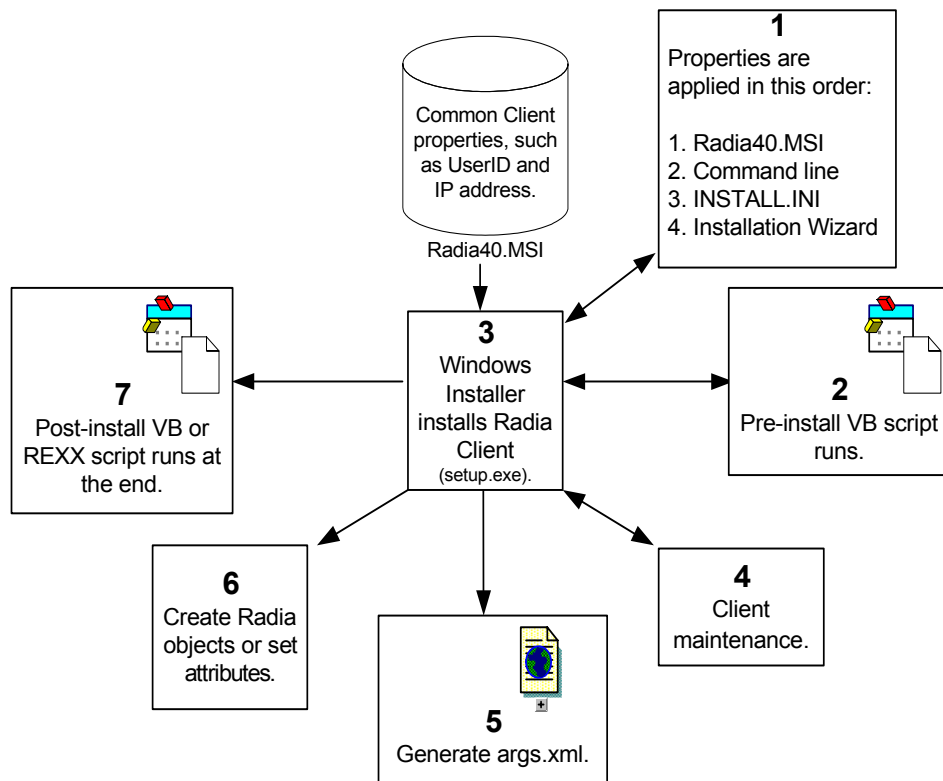
Figure 2.1 ~ Radia Client installation process.

Preparing INSTALL.INI

Create an installation file, INSTALL.INI. Use this file to:

■ Customize the installation.

■ Customize the Radia Software Manager.

■ Create or set attributes for Radia objects.

Below is a sample INSTALL.INI. A description of the three sections and the attributes in each of these sections follows.

Notes

A semi-colon (;) at the beginning of attributes, properties, and object names indicates that this item is commented out and will be ignored. If you want to specify a value for any of these items, delete the semi-colon (;) and then type the appropriate value.

Table 2.1 ~ Sample INSTALL.INI

```
[Properties]
;.MSI Property overrides (Novadigm properties have the "nvd" prefix)

;Uncomment and specify the installation directory to replace the default directory
;INSTALLDIR=

;Uncomment to prevent subscribers from using Add/Remove Programs in the Control Panel to
remove the Radia Client
;ARPNOREMOVE=1

;Uncomment and specify the features that you want to install
;ADDLOCAL=NVDINSTALLRAM,NVDINSTALLRSM,NVDINSTALLRIM,NVDINSTALLROM,NVDINSTALLPATCH,
NVDINSTALLRLAE

;Uncomment to show/hide panels in the installation wizard (defaults shown below)
;NVDENABLEUSER=Y
;NVDENABLEIP=Y
;NVDENABLEPROXY=N
;NVDENABLESHORTCUT=Y
;NVDSHORTCUT=Y
;NVDSTARTMENUICON=Y
;NVDSTARTWMICFGMGR=Y

;Uncomment and specify the location of maintenance files
;NVDMAINTDIR=

;Uncomment to allow notifies only from the local host.
;NVDLOCALNOTIFYONLY=Y

;Uncomment to start the System Tray automatically if RAM is selected during the client
install process.
;NVDRADTRAYSTART=Y

;Uncomment to disable "Allow service to interact with desktop" flag for Radia Services
;NVDNOTIFYINTERACT=N
;NVDREDIRECTORINTERACT=N

;NVDSCHEDULERINTERACT=N
```

Table 2.1 ~ Sample INSTALL.INI

```
;Uncomment and specify the file names of pre- and post-install custom action scripts
;NVDPRECAPATH=
;NVDPOSTCAPATH=

[Args]
; Tags to be placed into the Radia Software Manager ARGS.XML file
; A value of _NONE_ removes the tag from the file
;
;askconfirm=
;channelname=software
;dataurl=
;enterprisemanagement=
;identification=$USER
;log=connect.log
;logsize=
;logonpanel=
;managerurl=
;providername=Radia
;redirect=
;resolutionmanager=
;resolutionport=
;sslmanager=
;sslport=
;startdir=
;uioption=

[Objects]
; Set Radia object attribute values
; A value of _NONE_ will set the attribute to blank
;
;ZMASTER_ZDSTSOCK=3464
;ZMASTER_ZIPADDR=xxx.xxx.xxx.xxx
;ZMASTER_ZNTFPORT=3465
;ZMASTER_ZNTFYSEC=Y
;ZMASTER_ZTIMEO=240
;ZMASTER_ZTRACE=N
;ZMASTER_ZTRACEL=_NONE_
;ZMASTER_ZUSERID=
;ZMASTER_ZVRFYUID=N
;
;PROXYINF_USEPROXY=
;PROXYINF_DISCOVER=
;PROXYINF_PROXADDR=
;
; Uncomment to enable Client Operations Profile (COP)
;RADSETUP_COP=Y
; Uncomment to disable collection of AD information
;RADSETUP_ADINFO=N
; Uncomment to disable collection of NT Group information
 ;RADSETUP_ZGRPINFO=N
```

[PROPERTIES] Section of INSTALL.INI

Use the [PROPERTIES] section to modify Windows Installer properties or HP-specific properties to customize the behavior of the installation program. The values that you set in this section override the default values stored in the RADIA40.MSI database file.

Notes

All properties such as INSTALLDIR must be typed in all uppercase.

Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI

Argument	Description	Default Value*
	*These values are used even if the property is commented out.	
ADDLOCAL	Specify the features that you want to install on the local hard drive. You must be properly licensed to use these products. The features may be: • NVDINSTALLRAM = Radia Application Manager • NVDINSTALLRSM = Radia Software Manager • NVDINSTALLRIM = Radia Inventory Manager • NVDINSTALLROM = Radia OS Manager • NVDINSTALLPATCH = Radia Patch Manager • NVDINSTALLRLAE = Radia Local AIP Extension **Note:** This must be installed to use the Local AIP support for the Radia MSI Redirector feature described on page 412.	N/A
ARPNOREMOVE	Set ARPNOREMOVE to 1 to indicate that you want to disable the ability to remove the Radia Client from the computer using Add/Remove Programs in the Control Panel. **Note:** Setting ARPNOREMOVE to 0 will *not* disable this option due to a Windows Installer issue. If you want to allow your subscribers to remove the Radia Client from the computer using Add/Remove Programs in the Control Panel, place a semi-colon (;) in front of the ARPNOREMOVE argument in INSTALL.INI. For Windows 2000 or XP, the **Remove** button is disabled. For earlier operating systems, the Radia Client will not be listed in Add/Remove Programs in the Control Panel.	Subscribers can remove the Radia Client from the computer using Add/Remove Programs in the Control Panel.
INSTALLDIR	Specify the directory where you want to install the Radia Client. This value will be overridden if a new directory is specified in the **Destination Folder** window in the Radia 4.x Client Installation Wizard.	*SystemDrive*:\Program Files\Novadigm

Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI

Argument	Description	Default Value*
	* These values are used even if the property is commented out.	
NVDENABLEUSER	Indicate whether to show or hide the **Set User** window in the Installation Wizard. Y – Show the window. N – Hide the window. **Note**: If you hide the window, the **Create a Radia Software Manager icon on the desktop** check box will no longer be available to your subscribers. D – Show the window, but disable the **User Name** field. The **Create a Radia Software Manager icon on the desktop** check box is still available.	Y
NVDENABLEIP	Indicate whether to show or hide the **Radia Configuration Server** window in the Installation Wizard.	Y
NVDENABLEPROXY	Indicate whether to show or hide the **Proxy Information** window in the Installation Wizard. If you want to use a Radia Proxy Server during the Radia Client Connect, show this window. The information entered in the **Proxy Information** window is stored, by default, in the PROXYINF object in the client computer's IDMLIB directory.	N
NVDENABLESHORTCUT	Indicate whether to show the **Create a Radia Software Manager icon on the desktop** check box in the **Set User** window. Selecting this check box installs a desktop shortcut for the Radia Software Manager on the subscriber's desktop. **Reminder:** This property does not apply to computers running Windows 95 because the Radia Software Manager 4.x client is not available for computers running Windows 95, as it requires Microsoft's .NET runtime.	Y
NVDSHORTCUT	Indicate whether to install a desktop shortcut for the Radia Software Manager on the subscriber's computer. **Reminder:** This property does not apply to computers running Windows 95 because the Radia Software Manager 4.x client is not available for computers running Windows 95, as it requires Microsoft's .NET runtime.	Y

Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI

Argument	Description	Default Value*
	These values are used even if the property is commented out.	
NVDSTARTMENUICON	Indicate whether to install an icon in the Start Menu for the Radia Software Manager on the subscriber's computer. **Reminder:** This property does not apply to computers running Windows 95 because the Radia Software Manager 4.x client is not available for computers running Windows 95, as it requires Microsoft's .NET runtime.	Y
NVDSTARTWMICFGMGR	Indicates whether to install the shortcuts for WMI on Windows NT and 9x systems.	Y
NVDRAMSHORTCUT	Indicate whether to install a desktop shortcut for the Radia Application Manager on the subscriber's computer.	N
NVDRAMSTARTMENUSHORTCUT	Indicate whether to install an icon in the Start Menu for the Radia Application Manager on the subscriber's computer.	N
NVDRAMCONNECT	Specify a command line to run if a Radia Application Manager shortcut is created on the desktop or the Start Menu.	Blank
NVDMAINTDIR	Specify a directory that stores the Radia Client maintenance files. **Note:** Type a value only if you want to store maintenance files in a directory other than the MAINT subdirectory in the folder containing setup.exe. If files exist in this directory that are newer than the installation files, they will be copied into the Radia Client's IDMSYS directory.	MAINT subdirectory in the folder containing setup.exe
NVDLOCALNOTIFYONLY	If set to **Y**, the Client will allow Radia Notifies only from the local host.	N
NVDRADTRAYSTART	Set to **Y** to start the System Tray automatically, if the Radia Application Manager is selected during the Client installation process.	N
NVDNOTIFYINTERACT	Set to **Y** to enable the Radia Notify Daemon to interact with the desktop.	N
NVDREDIRECTORINTERACT	Set to **Y** to enable the Radia MSI Redirector to interact with the desktop.	N
NVDSCHEDULERINTERACT	Set to **Y** to enable the Radia Scheduler to interact with the desktop.	N

Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI

Argument	Description	Default Value*
	* These values are used even if the property is commented out.	
NVDPRECAPATH	Specify the fully qualified path and filename of a custom Visual Basic pre-install script. **Note:** New objects or properties must be defined in INSTALL.INI. You can use a pre-install script to override a value for the object or property, but if you attempt to specify a new object or property in the pre-install script, it will be ignored. For an example of a simple script, see *Using a Pre-Install Script* on page 79.	N/A
NVDPOSTCAPATH	Specify the fully qualified path and filename of a custom Visual Basic or REXX post-install script. For an example, see *Using a Post-Install Script* on page 81.	N/A

[ARGS] Section of INSTALL.INI

Use the [ARGS] section to control the behavior of the Radia Software Manager. The information in this section is used to build the Radia Software Manager arguments file, args.xml, which is stored in IDMLIB on the client computer. The default directory for IDMLIB is *SystemDrive*:\Program Files\Novadigm\Lib\.

Below is an example of args.xml.

```xml
<?xml version="1.0" ?>
<RADIA_ARGUMENTS>
<ARGUMENTS><CHANNELNAME>software</CHANNELNAME>
<IDENTIFICATION>jsmith</IDENTIFICATION>
<PROVIDERNAME>radia</PROVIDERNAME>
<RESOLUTIONMANAGER>10.10.10.1</RESOLUTIONMANAGER>
<LOG>connect.log</LOG>
<RESOLUTIONPORT>3464</RESOLUTIONPORT>
</ARGUMENTS>
</RADIA_ARGUMENTS>
```

Figure 2.2 ~ Sample args.xml.

Note

The XML tags (arguments) described in this section are not case-sensitive when you type them in INSTALL.INI. However, the arguments will automatically be converted to all uppercase in args.xml.

If you are using the Radia Application Manager, any of the parameters in the [ARGS] section can be added to the **radskman** command line.

Table 2.3 ~ [ARGS] Section of INSTALL.INI

Argument	Mandatory or Optional	Description	Default Value
askconfirm	Optional	Controls the display of a confirmation message to your subscribers. For example, some instances in which a confirmation message may display are: • A reboot is required. • There is insufficient disk space during deployment. • A data download is interrupted.	Y
channelname	Mandatory	The domain in the Radia Configuration Server Database from which applications are retrieved.	SOFTWARE

Table 2.3 ~ [ARGS] Section of INSTALL.INI

Argument	Mandatory or Optional	Description	Default Value
enterprisemanagement	Optional	*For use with the Radia Software Manager only.* Deploys mandatory applications from the Radia Software Manager. Set Enterprisemanagement=auto to deploy mandatory applications. Remove the Enterprisemanagement tag if you do not want to deploy mandatory applications.	N/A
identification	Optional	Identifies the client to the Radia Configuration Server by defining the value for the ZUSERID variable in the ZMASTER object. This value will be overridden if a different User Name is specified in the **Set User** window in the Radia 4.x Client Installation Wizard. If you do not want this value to be modified, set NVDENABLEUSER = N in the [PROPERTIES] section of INSTALL.INI. • **$MACHINE** – The Radia user ID is the name of the subscriber's computer. • **$USER** – The Radia user ID is the logon ID for the subscriber currently logged on. • **CUSTOM** – literal custom specification.	$USER
log	Optional	Specifies the name of the log stored in IDMLOG. IDMLOG is specified in NVD.INI. NVD.INI is stored in *SystemDrive*:\Program Files\Novadigm\LIB, by default.	Connect.log
logsize	Optional	Specifies the size of the log file in bytes. When the logsize is reached, a backup file (.BAK) is created. By default, this file is connect.bak. If a backup file already exists, it will be overwritten.	1000000
logonpanel	Optional	Controls the display of the logon panel.	Y
managerurl	Optional	Specifies the address of the Radia Configuration Server to be used for HTTP object transfer. The **managerurl** specification takes the form http://*hostname:port/nvdurl*.	N/A
providername	Mandatory	The name of the Radia Configuration Server, which is set during the standard installation. This is used to name the folder below the STARTDIR on the client computer. See *Startdir* on page 44 for more information.	Radia

Table 2.3 ~ [ARGS] Section of INSTALL.INI

Argument	Mandatory or Optional	Description	Default Value
redirect	Optional	*Used for the Radia Software Manager only.* Specifies an alternate start-up file (*filename*.xml), which can be accessed via a network path or URL. If the redirect tag is set in args.xml, the Radia Software Manager uses the properties specified in the alternate file. For example, you might set redirect to point to a start-up file on the network that is shared by all users.	N/A
resolutionmanager	Mandatory	Identifies the IP address for the Radia Configuration Server. You can also use the server name. This value will be overridden if a different IP address is specified in the **Radia Configuration Server** window in the Radia 4.x Client Installation Wizard. If you do not want this value to be modified, set NVDENABLEIP = N in the [PROPERTIES] section of the INSTALL.INI.	N/A
resolutionport	Mandatory	Identifies the port for the Radia Configuration Server. This value will be overridden if a different port is specified in the **Radia Configuration Server** window in the Radia 4.x Client Installation Wizard. If you do not want this value to be modified, set NVDENABLEIP = N in the [PROPERTIES] section of the INSTALL.INI.	N/A
sslmanager	Optional	Specifies the address of the Radia Configuration Server to be used for SSL communications. If you want self-maintenance to use SSL communications, concatenate **::SM** to the end of the specified IP address or host name. For example, sslmanager=*hostname*::SM **Warning:** Use the **::SM** switch with the following caveat in mind; the file (cacert.pem) that contains the CA root certificates cannot be maintained. If the corresponding CA root certificate for the certificate in use by the Radia Configuration Server should ever become expired, revoked, or corrupt, it will result in disabling SSL communications to the Radia Configuration Server.	N/A
sslport	Optional	Specifies the TCP/IP port that the SSL manager is listening on. This is normally 443. The **sslport** specification takes the form sslport=*port*.	N/A

Table 2.3 ~ [ARGS] Section of INSTALL.INI

Argument	Mandatory or Optional	Description	Default Value
startdir	Optional	Specifies the starting IDMLIB directory (by default, *SystemDrive*:\Program Files\Novadigm\Lib\). • **$MACHINE** – Uses the current computer name. • **$USER** – Uses the subscriber who is logged on. • **CUSTOM** – Literal custom specification. Type startdir=*foldername*. If the folder name contains embedded spaces, enclose the entire name in double quotes. **Note**: You might want to do this to distribute a set of common applications to be shared by all subscribers of a multi-user computer.	$USER
uioption	Optional	Controls the display of the status window.	N

[OBJECTS] Section of INSTALL.INI

Use the [OBJECTS] section to specify Radia objects to be created on the client computer and to set their default values. The format is *clientobject_attbribute*. For example, if you wanted to set the IP address for your Radia Configuration Server, set ZMASTER_ZIPADDR.

Table 2.4 ~ [OBJECTS] Section of INSTALL.INI		
Argument	**Description**	**Default Value**
ZMASTER_ZDSTSOCK	The port setting for the Radia Configuration Server.	3464
ZMASTER_ZIPADDR	The IP address for the Radia Configuration Server.	N/A
ZMASTER_ZNTFPORT	The port that the Radia Client's notify daemon is 'listening' on.	3465
ZMASTER_ZNTFYSEC	This attribute allows a Notify operation to execute programs only from the IDMSYS directory. This is used for security during Notify operations.	Y
ZMASTER_ZTIMEO	Amount of time that the Radia Client will wait for a response from the Radia Configuration Server before the Radia Client program is deactivated (timed out). Numeric value between 0 and 3200, in seconds.	240
ZMASTER_ZTRACE	Enables log to include communications buffer information; also generates unique logs for create methods. **Y** Turns on Communication and Client Method Tracing. **S** Turns on Communication 'summary' information; Client Method Tracing is *not* enabled. **N** Communication Tracing and Client Method Tracing are disabled.	N
ZMASTER_ZTRACEL	Level of tracing generated in the Radia Client log files. 0 to 999, where 0=minimal tracing, 40=acceptable for most activity, and 999=maximum.	040
ZMASTER_ZUSERID	The subscriber's user ID.	The name of the user currently logged on to the computer.
ZMASTER_ZVRFYUID	Verify the user ID sent by the Radia Configuration Server's Notify command with the ZUSERID field found in the Radia Client's ZMASTER object.	N
PROXYINF_USEPROXY	Indicates whether you use a proxy server to connect to the Radia Configuration Server.	N
PROXYINF_DISCOVER	*For use with Microsoft's Internet Explorer.* Set to the proxy address and port that your Web browser is using.	N
PROXYINF_PROXADDR	The IP address and port number of your proxy server.	xxx.xxx.xxx.xxx:1080
RADSETUP_COP	Set this to Y to enable Radia Client Operations Profiles.	N

Installing the Radia 4.x Clients

Before installing the Radia Clients, you must decide whether to initiate the installation from:

- a command line that initiates the installation from a Web page, FTP site, mapped drive, CD-ROM, or e-mail. See *Initiating the Radia Client Installation from a Command Line* below.
- a logon script. See *Initiating the Radia Client Installation from a Logon Script* on page 49.
- the Radia Management Portal. This is recommended for mass rollouts in a Windows NT, 2000, or XP environment. See the *Radia Management Portal Guide* for more information.

After initiating the installation, the Radia Client 4.x Installation Wizard runs. This section describes some of the ways that you can initiate the Radia Client installation, and then describes the standard Radia Client 4.x Installation Wizard.

Initiating the Radia Client Installation from a Command Line

Before performing an installation from a command line, you must determine:

- how you are going to make the Radia Client installation program available to your subscribers. You can do this via a Web page, FTP site, mapped drive, CD-ROM, or e-mail.
- what Radia Client features (Radia Application Manager, Radia Software Manager, and Radia Inventory Manager) you want to install and pass any additional arguments to the command line.

EXAMPLE

The following is an example of a command line that will install the Radia Software Manager silently and create a detailed Windows Installer log.

```
SETUP.EXE ADDLOCAL=NVDINSTALLRSM /qn /L*v C:\Novadigm\install.log
```

The arguments in this command line, and others, are described in Table 2.5 below and Table 2.6 on page 48.

Specifying the Radia Client Features to Install

To specify the features that you want to install, use the appropriate feature state argument, such as ADDLOCAL, and specify the features that you want to install.

Table 2.5 ~ Radia Client Feature State Arguments	
Specify the following arguments	**To set the feature state**
ADDLOCAL	Type a comma-delimited list of features that you want set to "Will be installed on local hard drive."

Table 2.5 ~ Radia Client Feature State Arguments

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

When specifying features on the command line, reference the Radia Client features as follows:

- NVDINSTALLRAM
  Installs the Radia Application Manager.

- NVDINSTALLRSM
  Installs the Radia Software Manager.

- NVDINSTALLRIM
  Installs the Radia Inventory Manager.

- NVDINSTALLROM
  Installs the Radia OS Manager.

- NVDINSTALLPATCH
  Installs the Radia Patch Manager.

For example, if you want to install the Radia Application Manager and the Radia Software Manager to the computer, the command line might be:

    SETUP.EXE ADDLOCAL= NVDINSTALLRAM,NVDINSTALLRSM

Additional Command Line Arguments

Some additional arguments that you can pass to the installation program on the command line are described Table 2.6 below.

Table 2.6 ~ Command Line Arguments	
Argument	**Description**
/qn	Performs a silent installation. **Note**: A silent installation is one that takes place without a user interface. Throughout our literature, the terms "silent installation," "quiet installation," and "unattended installation" are all used to refer to the same process.
/qb	Displays the progress bar only during the installation.
/L*v *drive*:*install.log*	Creates a detailed Windows Installer log. **Note:** Using this option may impact the performance of the installation.
/a TARGETDIR= *drive*:*targetdirectory*	Creates a Windows Installer AIP in the specified target directory. **Note:** A Windows Installer Administrative Installation Point (AIP) is also known as an Administrative Control Point (ACP). The target directory contains RADIA.MSI, the installation folders, setup.exe, and any files (such as INSTALL.INI or Visual Basic scripts) stored in the same directory as setup.exe. Once you have created the AIP, you can run setup.exe and pass the appropriate command line parameters. This starts the Windows Installer and passes the specified parameters to it.
NVDINIFILE=*path|INIfilename*	If you want to rename the installation INI file, pass this parameter to the command line. Be sure to include the fully qualified path. By default, the installation program refers to INSTALL.INI located in the current directory.

If you initiate a Radia Client installation with a command line that does not contain the silent installation arguments (/qn), the Radia Client Installation Wizard opens. See *Using the Radia Client Installation Wizard* on page 51 for more information.

Initiating the Radia Client Installation from a Logon Script

You can use a logon script on an NT, Windows 2000, Windows Server 2003, XP, or Netware client to automate installation of the Radia Clients.

Important Note

To install the Radia Clients automatically on subscriber's NT 4.0, Windows 2000 Professional, or XP computer, subscribers *must* have administrator rights on their local computers, and a domain controller must authenticate each subscriber's logon.

The following is an example of code that can be added to your logon script that installs the Radia 4.x Clients. If the Radia Clients are *not* already installed when the subscriber logs on to the server, this logon script runs the Radia 4.x Client installation program.

```
:begin
@echo off
if exist C:\progra~1\Novadigm\LIB\zmaster.edm goto skipinst
   start setup.exe /qn
:skipinst
```

Figure 2.3 ~ Sample logon script.

```
if exist C:\progra~1\Novadigm\lib\zmaster.edm goto skipinst
```

To determine if the Radia Clients already exist, the script checks to see if the ZMASTER object (ZMASTER.EDM) exists in its default location on the local computer.

Note

The ZMASTER object begins the resolution process and is the first object exchanged during the Client Connect process.

If ZMASTER.EDM *exists*, the script skips the installation. If ZMASTER.EDM *does not* exist, the Radia 4.x Client installation program launches.

```
start setup.exe /qn
```

This command line performs a silent installation of the Radia Clients.

Note

Modify this script to reflect your organization's needs.

If you initiate a Radia Client installation using a command line that does not contain the silent installation arguments, the Radia Client Installation Wizard opens. See *Using the Radia Client Installation Wizard* on page 51 for more information.

Using the Radia Client Installation Wizard

If you initiate a Radia Client installation without the arguments for a silent installation, the Radia Client Installation Wizard opens. The following steps describe the standard installation procedure. These steps may vary based on INSTALL.INI or any arguments passed when running the installation.

To install Radia Clients using the Installation Wizard

1. From the folder containing the Radia Client installation files, run **setup.exe**.

> **Reminder**
>
> You can initiate setup.exe from a command line, logon script, or from the Radia Management Portal. Go to the beginning of this chapter for more information.

The **Radia Client Installation Wizard** opens.

Figure 2.4 ~ Welcome window.

2. Click **Next**.

The License Agreement window opens.

Figure 2.5 ~ Read the License Agreement.

3. After reading and accepting the license agreement, click **Next**.

The **Destination Folder** window opens. The default location for the Radia Clients is *SystemDrive*:**\Program Files\Novadigm**.
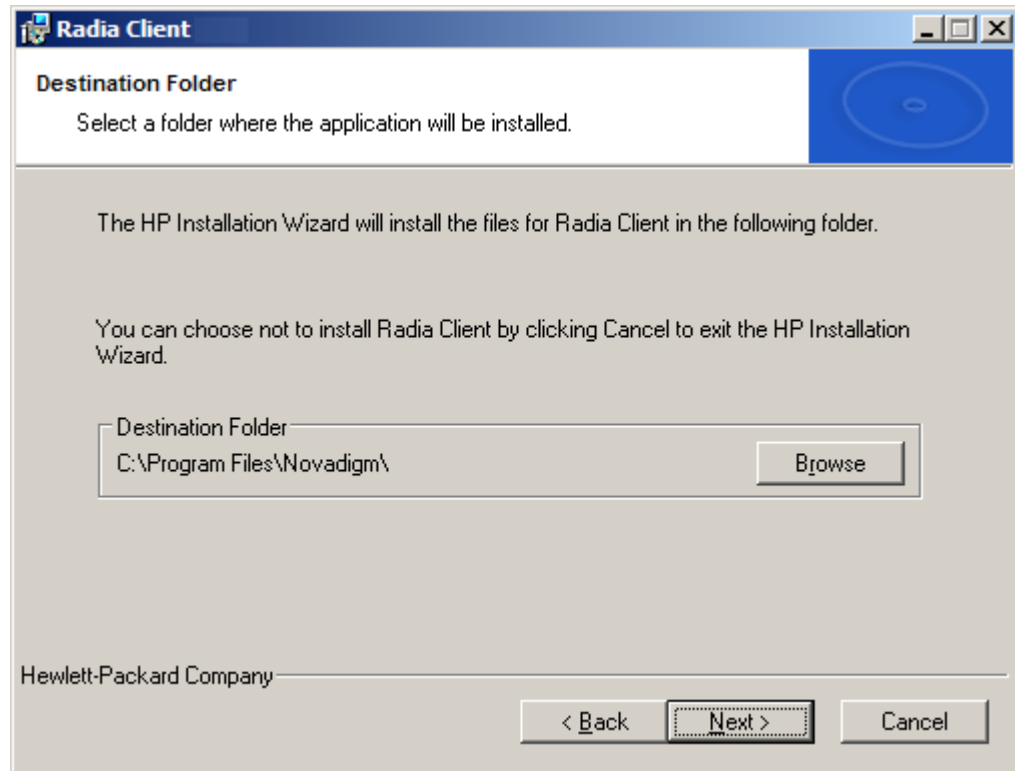
Figure 2.6 ~ Destination Folder window.

If you want to select a different destination for the Radia Client, click **Browse** and then navigate to the appropriate destination folder. This overrides the value set for INSTALLDIR in INSTALL.INI.

4. Click **OK** to continue.

5. Click **Next**.

The **Set User** window opens.
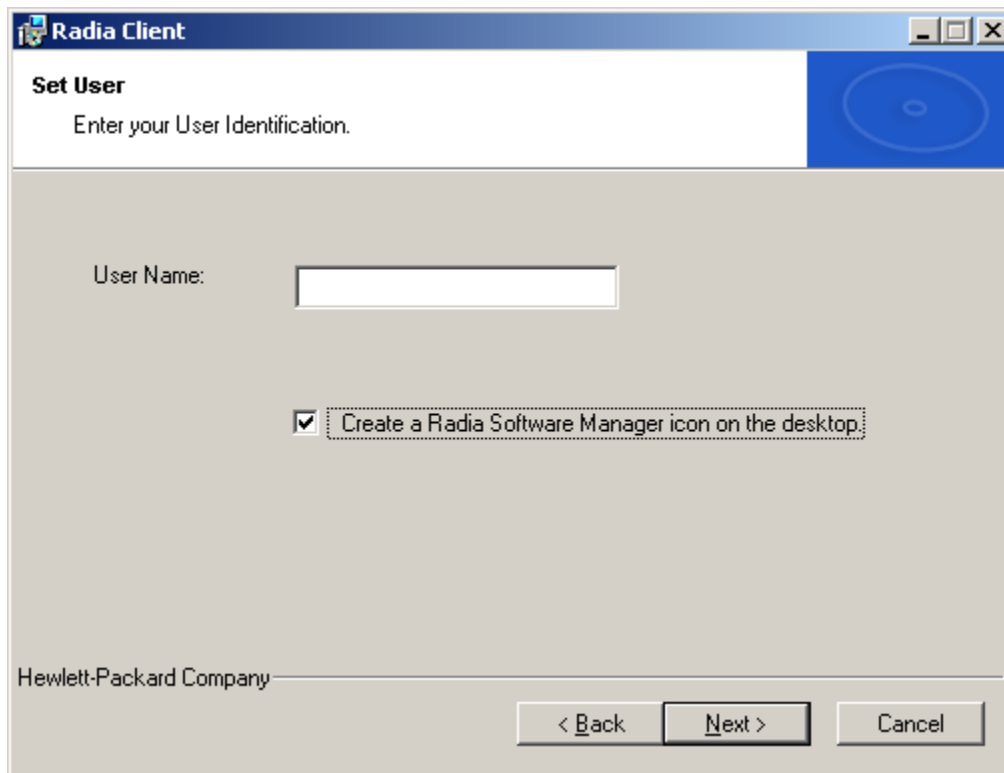
Figure 2.7 ~ Set User window.

6. In the **User Name** text box, type the name of the subscriber for whom you are installing the Radia Clients. This overrides the value set for IDENTIFICATION in INSTALL.INI.

7. Select the **Create a Radia Software Manager icon on the desktop** check box if necessary.

> **Note**
>
> This check box is not available in Windows 98. By default NVDSHORTCUT=Y, which means a shortcut will appear on the subscriber's desktop. Use NVDSHORTCUT=N to hide the shortcut. See the description of NVDSHORTCUT in *Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI* on page 37.

8. Click **Next**.

   The **Radia Configuration Server** window opens.
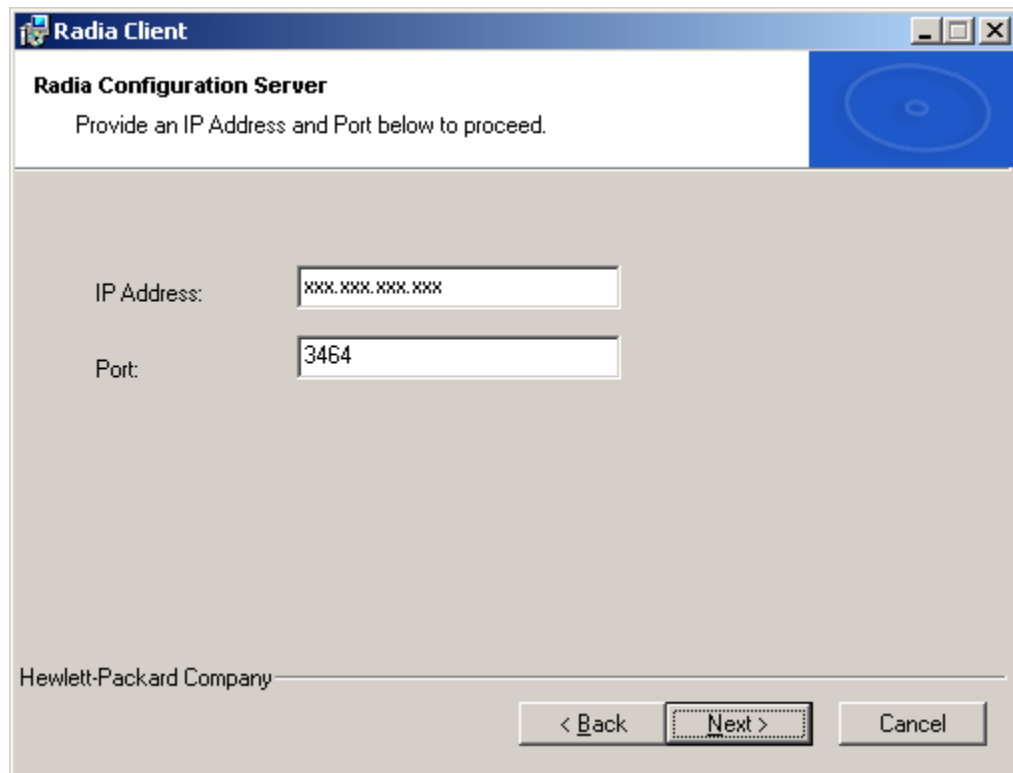
Figure 2.8 ~ Radia Configuration Server.

9. In the **IP Address** text box, type the IP address for the Radia Configuration Server. This overrides the value set for RESOLUTIONMANAGER in INSTALL.INI.

10. In the **Port** text box, type the port number. This overrides the value set for RESOLUTIONPORT in INSTALL.INI.

11. Click **Next**.

    The **Select Features** window opens.
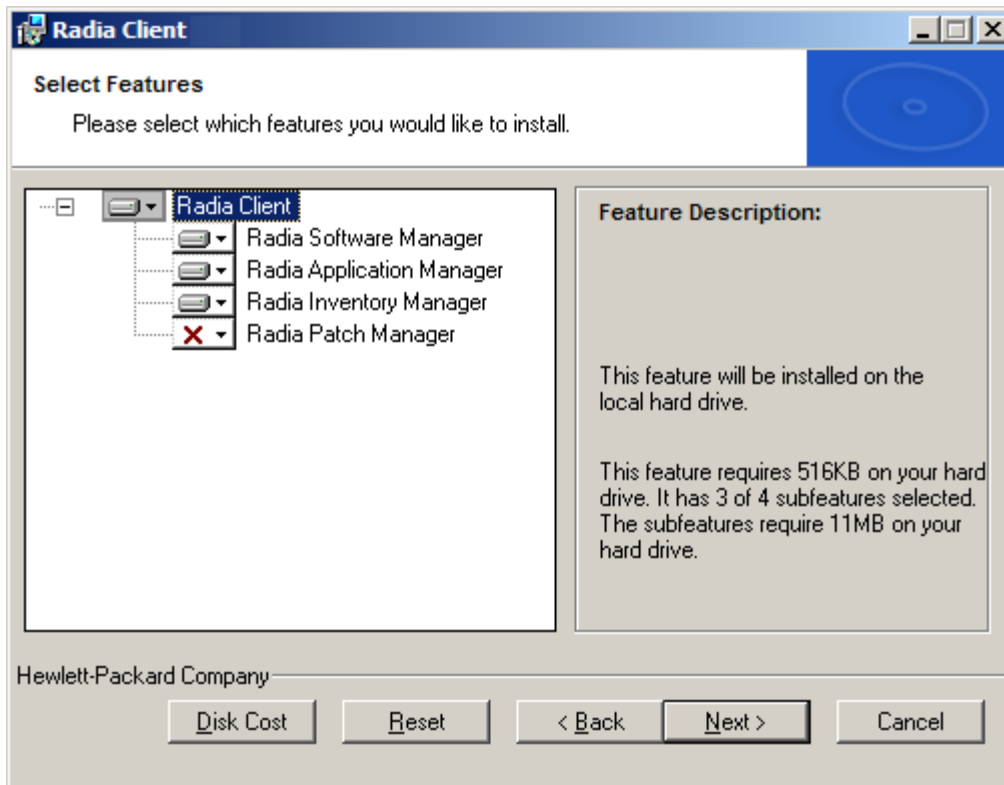
Figure 2.9 ~ Select Features window.

> **Note**
>
> The Radia Software Manager Client is not available for computers running Windows 95. If you use this installation program, the Radia Software Manager will not be an option.

12. Click ⬛▾ to select the features that you want to install.

Each time you click ⬛▾, a shortcut menu for that feature opens (see *Figure 2.10 ~ Feature set installation options* on page 58).

> **Note**
>
> Only install the Radia Clients for which you are licensed.
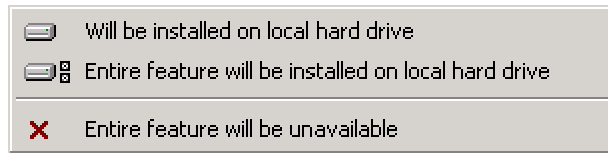
Figure 2.10 ~ Feature set installation options.

13. From the shortcut menu, select an installation option. These options are described in Table 2.7 below.

Table 2.7 ~ Feature Settings for Radia Clients	
Option	**Description**
Will be installed on local hard drive	Installs the top-level feature on the local hard drive, but not any sub-features listed below.
Entire feature will be installed on local hard drive	Installs the entire feature, including any sub-features listed below. **Note:** In this installation program, selecting this option or the "Will be installed on local hard drive" option for the Radia Software Manager, Radia Application Manager, or Radia Inventory Manager results in the same installation because these features do not contain sub-features. **Example:** If you selected this option for the Radia Client feature in the **Select Features** window, all of the Radia Clients would be installed on the local hard drive.
Entire feature will be unavailable	The feature will not be installed. If previously installed this feature will be removed.

Tips

· If you want to set the same options for all of the features, you can click **Radia Client** and select the appropriate option to apply the setting to all features.

· Click **Disk Cost** to see an overview of the disk space needed for the installation.

14. Click **Next**.

If .NET is not installed on the client computer and you have chosen to install the Radia Software Manager, .NET will be installed during the Radia Client installation. However, if you copied the installation program to your computer and did not include the **\DotNet** folder, the **DotNet Settings** message in Figure 2.11 on page 59 opens.
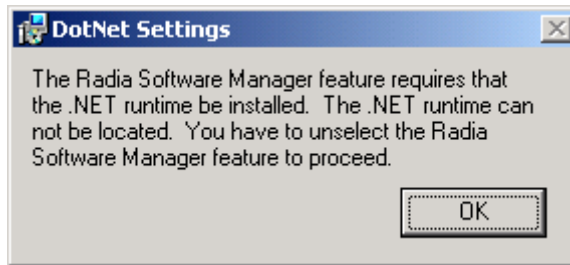
Figure 2.11 ~ DotNet Settings message.

15. Click **OK**.

If you do not want to install the Radia Software Manager, you can set this feature to be unavailable and then proceed. If you want to install the Radia Software Manager, you must copy the **\DotNet** folder to your computer. Be sure to copy this folder to the same directory where setup.exe is stored. Then, run setup.exe again.

16. If necessary, click **Next** again.

If .NET is not already installed on the computer, the **.NET Installation** window opens.
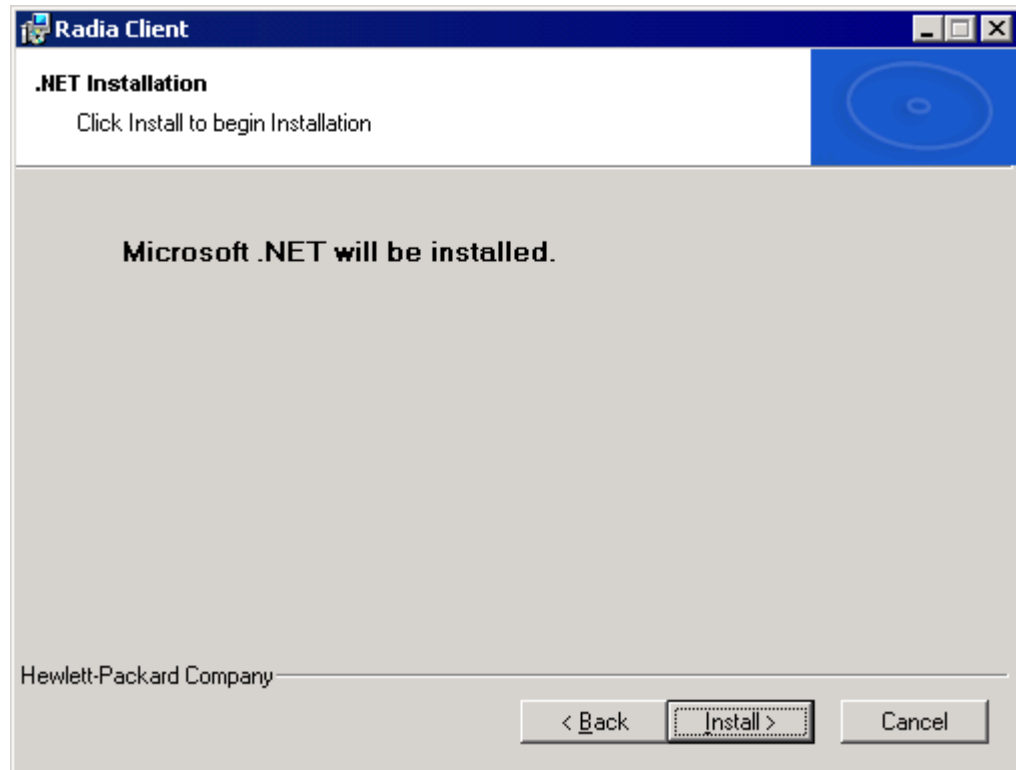
Figure 2.12 ~ .NET Installation window.

17. Click **Next**.

Caution
If you have installed .NET Beta, be sure to remove it before installing .NET.

The **Ready to Install the Application** window opens.
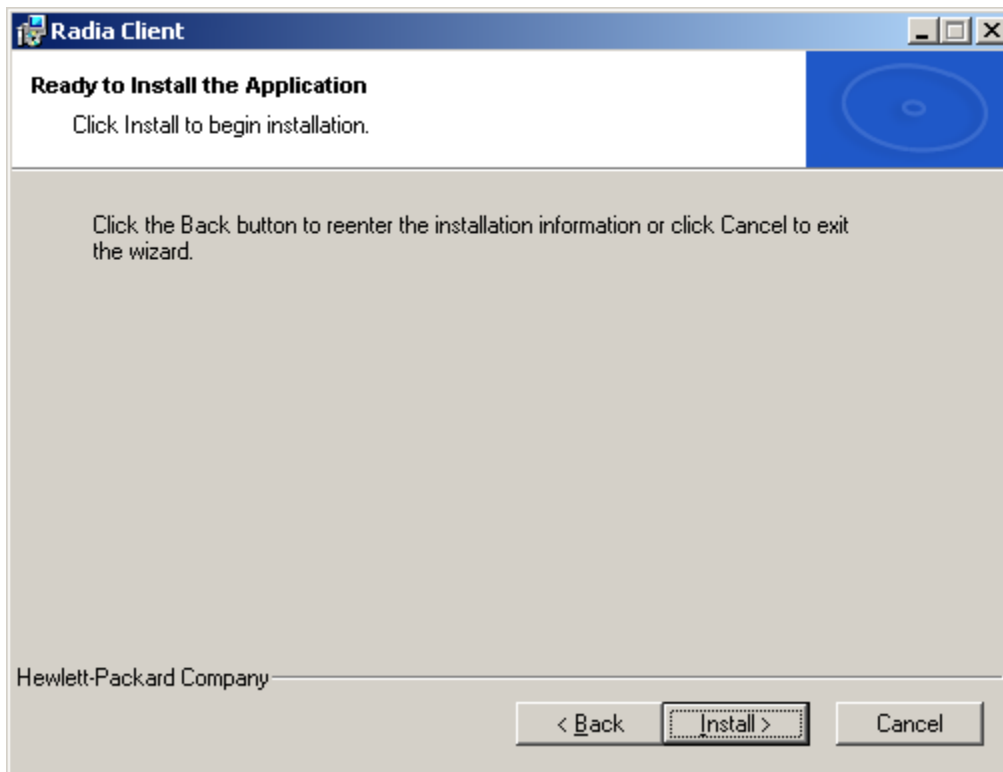
Figure 2.13 ~ Ready to Install the Application window.

18. Click **Install** to begin the installation.

If necessary, the **.NET Framework Setup** wizard opens. Follow the prompts to install .NET on the client computer. After .NET is successfully installed, the Radia Client installation begins.

When the installation is done, the **Radia Client has been successfully installed** window (Figure 2.14 below) opens.
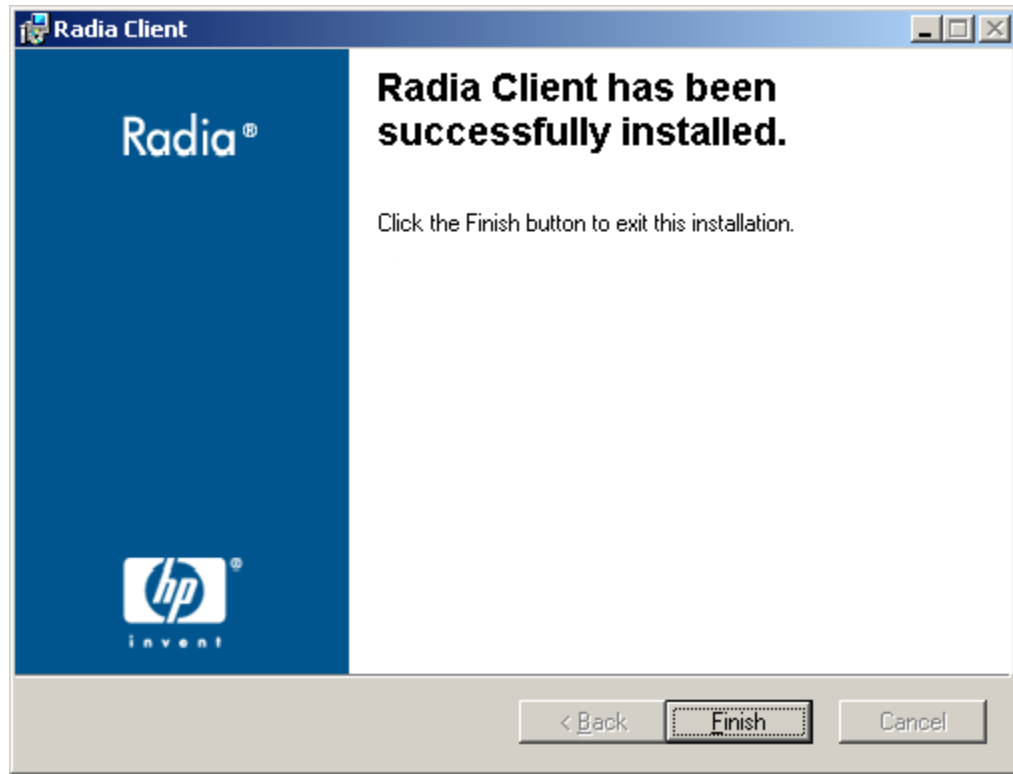
Figure 2.14 ~ Successful installation window.

19. Click **Finish** to close the Installation Wizard.

Removing the Radia Clients

The Windows Installer installation program offers the ability to remove your Radia 4.x Clients. This section describes how to remove the Radia Client using the Installation Wizard and using a command line.

Using the Installation Wizard to Remove Radia Clients

This section describes how to remove the Radia Client using the Installation Wizard.

> **Note**
>
> To remove specific features of the Radia Client, use the Modify option on the Application Maintenance window. This is discussed in *Modifying the Radia Client Installation* on page 73.

To remove Radia Clients using the Installation Wizard

1. From the folder containing the Radia Client installation files, double-click **setup.exe**. The **Application Maintenance** window opens.
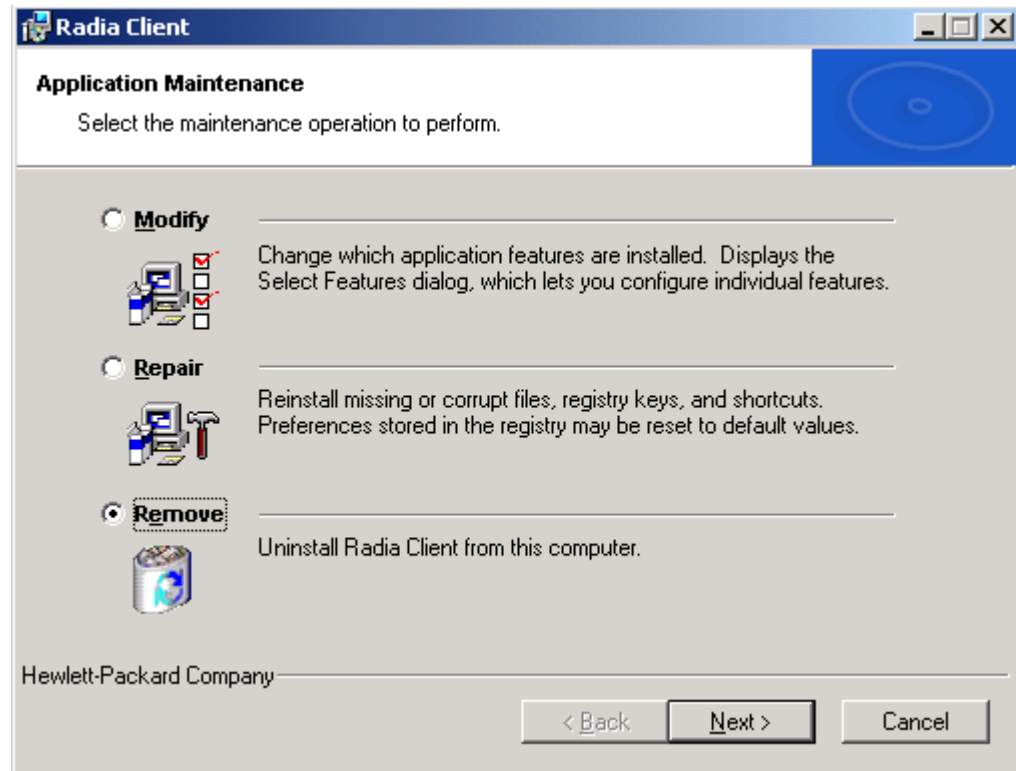
Figure 2.15 ~ Application Maintenance window.

2. Select the **Remove** option.

3. Click **Next**.

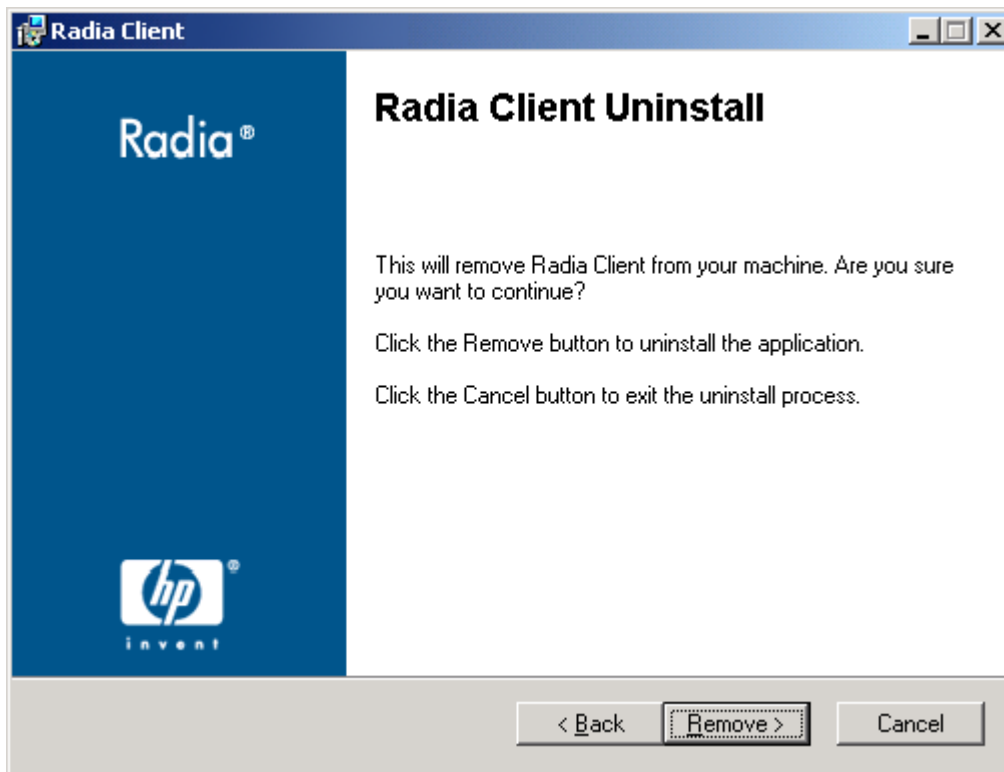The **Radia Client Uninstall** window opens.

Figure 2.16 ~ Radia Client Uninstall window.

4. Click **Remove.**

The files for all Radia Clients are removed from the computer.

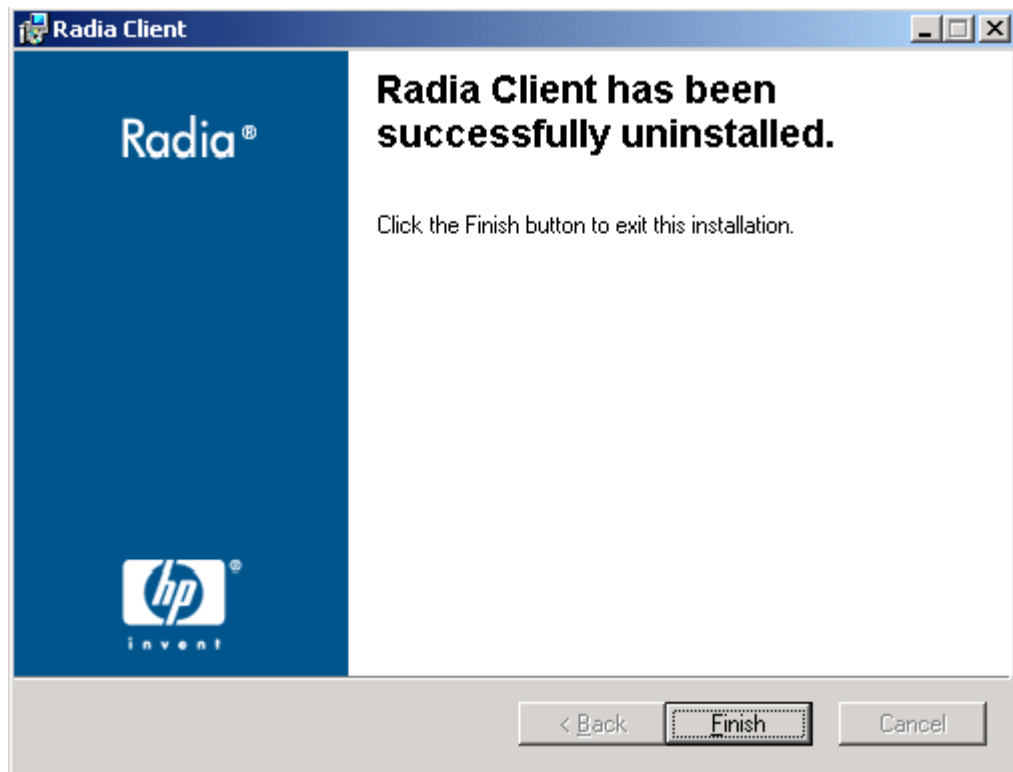The **Radia Client has been successfully uninstalled** window opens.

Figure 2.17 ~ Successful uninstallation window.

5. Click **Finish**.

Using a Command Line to Remove Radia Clients

This section describes how to remove Radia Clients using a command line.

To remove Radia Clients using a command line

■ From the folder containing the Radia Client installation files, type the following command line:

```
SETUP.EXE REMOVE=ALL
```

See *Initiating the Radia Client Installation from a Command Line* on page 46 for additional arguments.

OR

If you would like to remove a single Radia Client, type a comma-delimited list of the features that you want to remove on the command line.

Note
Reference the features for the Radia 4.x Client installation as follows: ▪ Radia Application Manager NVDINSTALLRAM ▪ Radia Software Manager NVDINSTALLRSM ▪ Radia Inventory Manager NVDINSTALLRIM ▪ Radia OS Manager NVDINSTALLROM ▪ Radia Patch Manager NVDINSTALLPATCH

EXAMPLE

If you want to silently remove the Radia Software Manager and Radia Application Manager, type:

```
SETUP.EXE REMOVE=NVDINSTALLRSM,NVDINSTALLRAM /qn
```

Note
This only removes the features—not the entire product. Therefore, if you use the REMOVE argument and type each of the feature names, the core product will still be stored on your computer.

Repairing the Radia Clients

The Windows Installer installation program offers the ability to repair your Radia 4.x Clients. For example, if you have a missing Radia Client module, you can use this tool to repair the installation. This tool will not overwrite modules that exist on the client computer if they are newer than the ones provided with the installation.

This section describes how to repair Radia Clients using the Installation Wizard and using a command line.

Using the Installation Wizard to Repair Radia Clients

This section describes how to repair Radia Clients using the Installation Wizard.

To repair Radia Clients using the Installation Wizard

1. From the folder containing the Radia Client installation files, double-click **setup.exe**.

The **Application Maintenance** window opens.
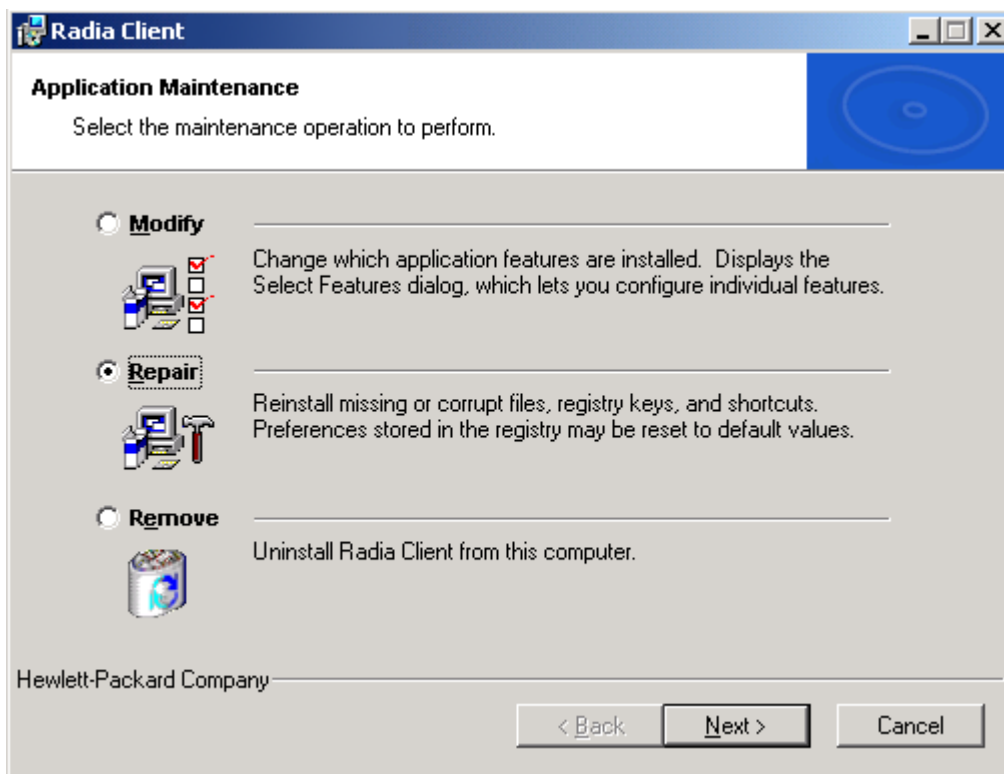
Figure 2.18 ~ Application Maintenance window.

2. Select the **Repair** option.
3. Click **Next**.

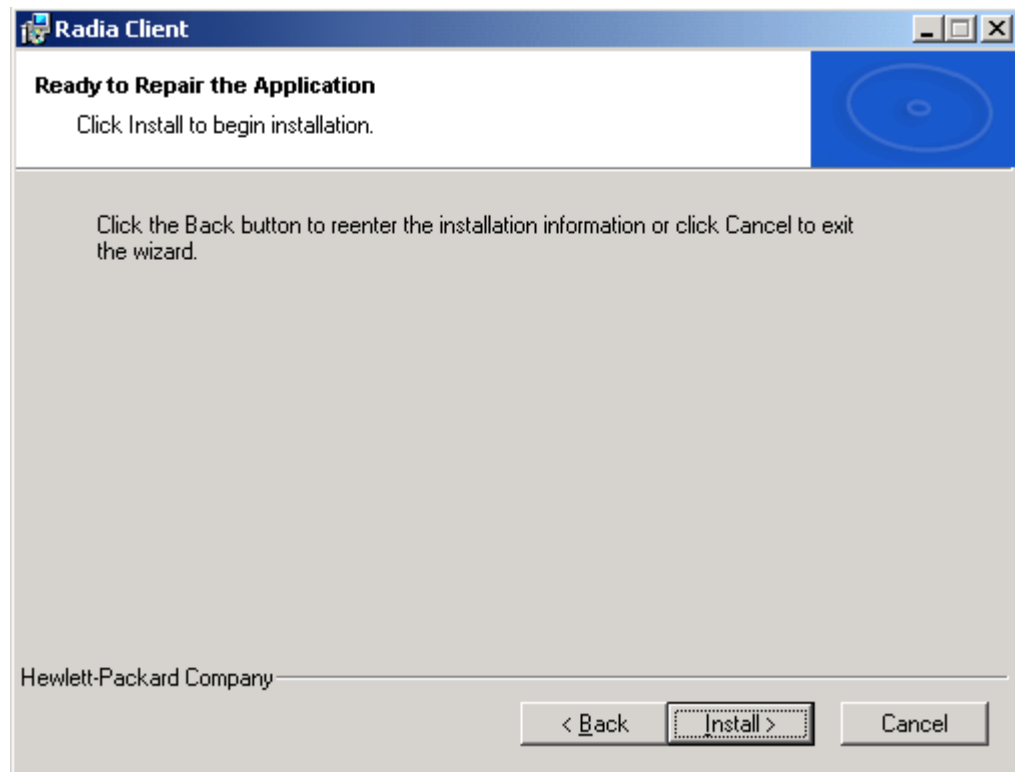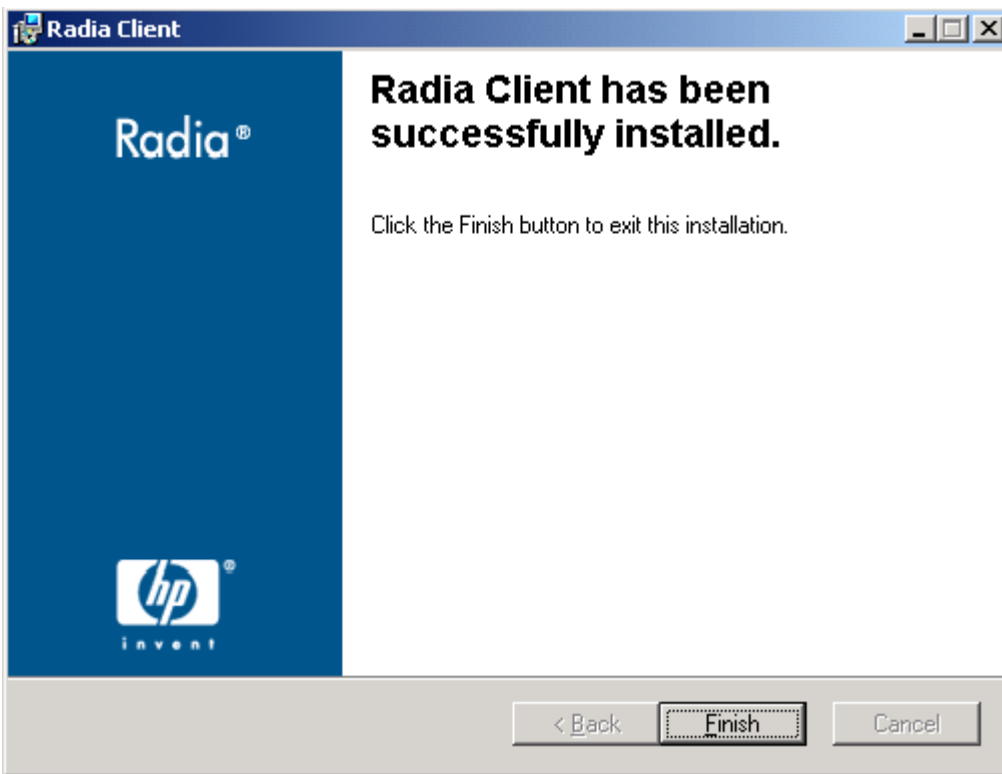The **Ready to Repair the Application** window opens.

Figure 2.19 ~ Ready to Repair the Application window.

4. Click **Next**.

When the repair is done, the **Radia Client has been successfully installed** window opens.

Figure 2.20 ~ Successful installation window.

5. Click **Finish**.

Using a Command Line to Repair Radia Clients

This section describes how to repair Radia Clients using a command line.

To repair Radia Clients using a command line

■ From the folder containing the Radia Client installation files, type the following command line:

```
msiexec /f RADIA40.MSI
```

> **Note**
>
> Additional parameters can be used with this command line. For more information, see your Windows Installer documentation.

Modifying the Radia Client Installation

The Windows Installer installation program offers the ability to modify your Radia 4.x Client installation by adding or removing individual features. This section describes how to modify the installation of Radia Clients using the Installation Wizard and using a command line.

Using the Installation Wizard to Modify the Installation of Radia Clients

This section describes how to modify the installation of Radia Clients using the Installation Wizard.

To modify the installation of Radia Clients using the Installation Wizard

1. From the folder containing the Radia Client installation files, double-click **setup.exe**.

The **Application Maintenance** window opens.
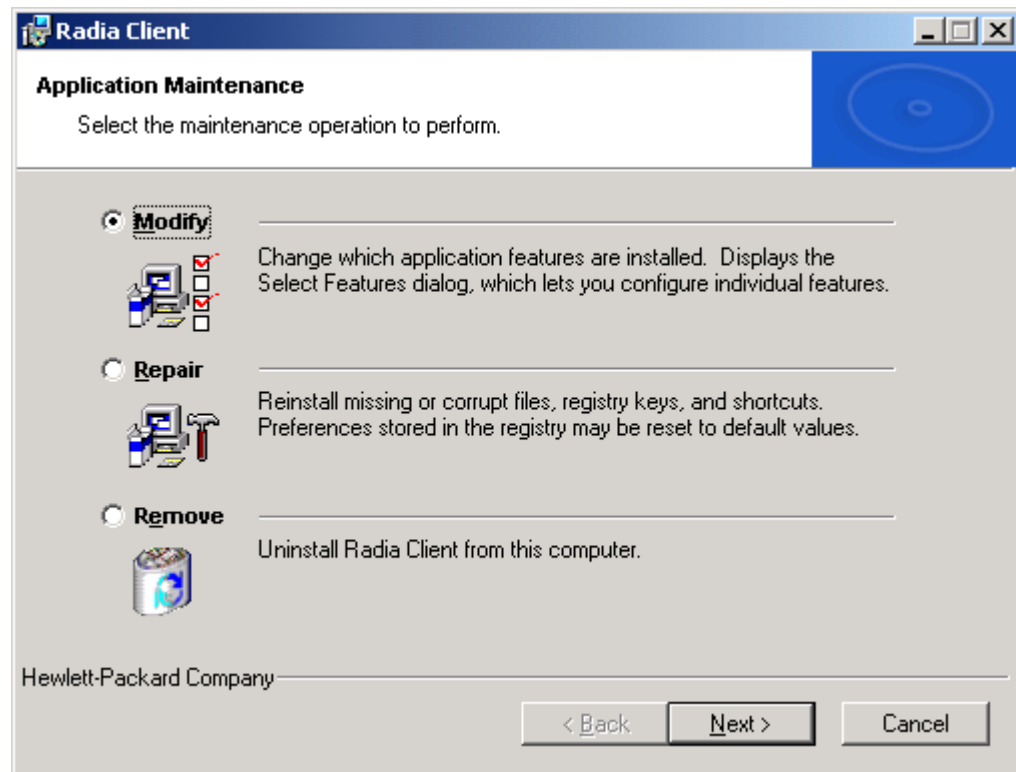
Figure 2.21 ~ Application Maintenance window.

2. Select the **Modify** option.

3. Click **Next**.

The **Select Features** window opens. See *Using the Radia Client Installation Wizard* on page 51 for information about how to use this window.
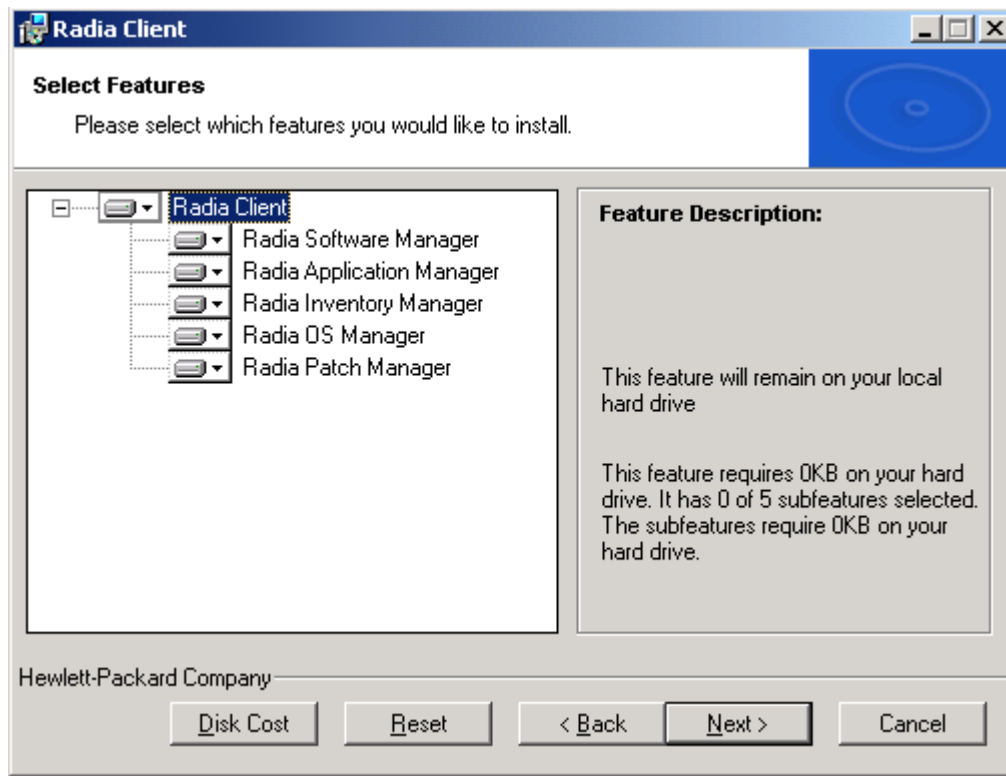

Figure 2.22 ~ Select Features window.

4. Click **Next**.
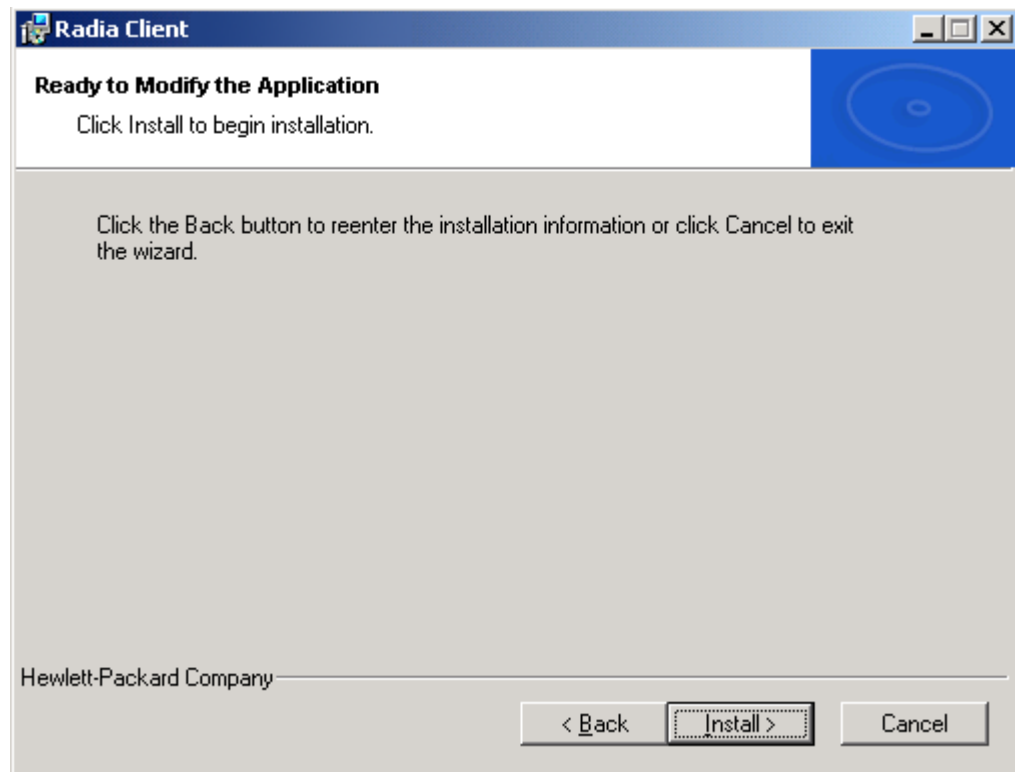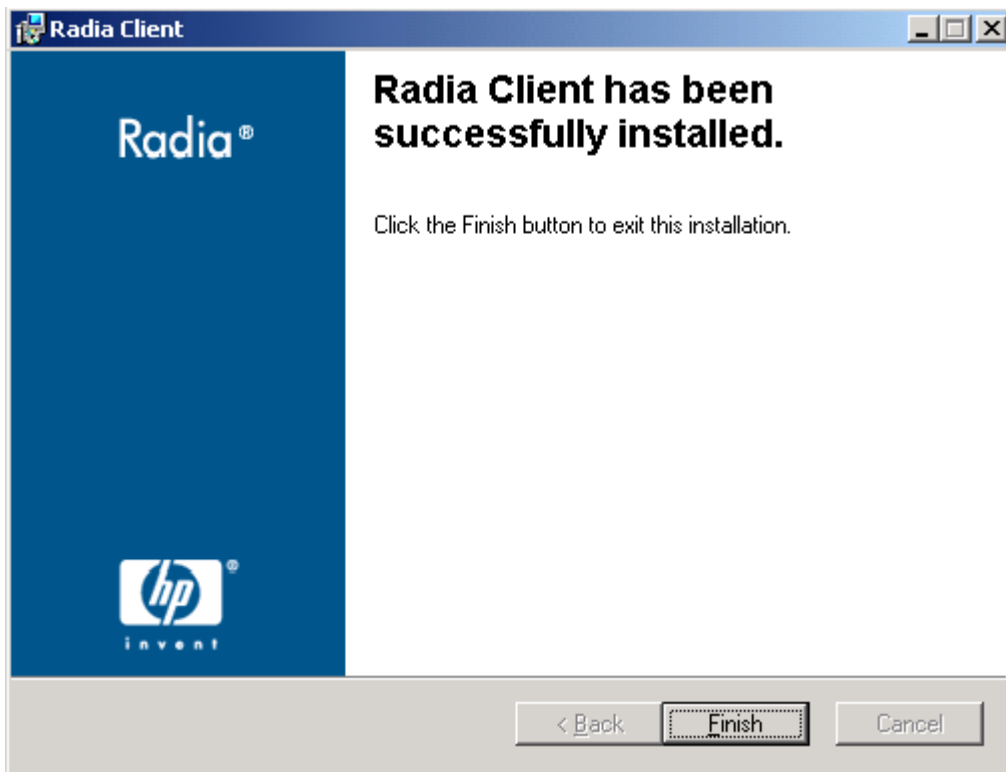
The **Ready to Modify the Application** window opens.

Figure 2.23 ~ Ready to Modify the Application window.

5. Click **Next**.

The **Radia Client has been successfully installed** window opens.

Figure 2.24 ~ Successful Installation window.

6. Click **Finish** to close the installation program.

Using a Command Line to Modify the Installation of Radia Clients

This section describes how to modify the installation of Radia Clients using a command line.

To modify the installation of Radia Clients using a command line

■ From the folder containing the Radia Client installation files, type the following command line:

```
SETUP.EXE FeatureStateArgument=feature1,feature2
```

Table 2.8 ~ Radia Client Feature State Arguments	
Specify the following arguments	**To set the feature state**
ADDLOCAL	Type a comma-delimited list of features that you want set to "Will be installed on local hard drive."
REMOVE	Type a comma-delimited list of features that you want set to "Entire feature will be unavailable."
	This only removes the features—not the entire product. Therefore, if you use the REMOVE property and type each of the feature names, the core product will still be stored on your computer.
	If you want to remove the entire product, type REMOVE=ALL.

Note

Reference the features for the Radia 4.x Client installation as follows:

- Radia Application Manager NVDINSTALLRAM
- Radia Software Manager NVDINSTALLRSM
- Radia Inventory Manager NVDINSTALLRIM
- Radia OS Manager NVDINSTALLROM
- Radia Patch Manager NVDINSTALLPATCH

EXAMPLE

If you want to install the Radia Software Manager, and to make the Radia Inventory Manager and Radia Application Manager unavailable, you would use the following command line:

```
SETUP.EXE ADDLOCAL=NVDINSTALLRSM REMOVE=NVDINSTALLRIM,NVDINSTALLRAM
```

See *Initiating the Radia Client Installation from a Command Line* on page 46 for additional arguments.

Using a Pre-Install Script

Use Visual Basic scripts to customize MSI properties that affect the installation. The following is a very simple Visual Basic script, which is intended to be an example only. This script is available in the **\win32\samples** directory on the Radia Client CD-ROM.

Caution
Be sure to use the NVDPRECAPATH argument to specify the fully qualified path and filename of a custom Visual Basic pre-install script in INSTALL.INI or on the command line. See the description of NVDPRECAPATH in *Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI* on page 37.

```
' The following sample demonstrates fetching an MSI property, then setting the
same property.
' The property values are displayed in message boxes for debugging purposes.
Option Explicit
msgbox Session.Property("ALLUSERS")
Session.Property("ALLUSERS") = "1"
msgbox Session.Property("ALLUSERS")
```

Figure 2.25 ~ Sample pre-install script.

You can use a pre-install script to override the property settings of the arguments that control the behavior of the Radia Software Manager, such as those in the [ARGS] section of INSTALL.INI, as well as the attribute values for Radia objects, such as those specified in the [OBJECTS] section of INSTALL.INI.

Caution
New objects or properties must be defined in INSTALL.INI.
You can use a pre-install script to override a value for the object or property, but if you attempt to specify a new object or property in the pre-install script, it will be ignored.

To override property settings or attributes for objects

Caution
Be sure to type the name of the property or the object and its attribute such as NVDOBJZMASTER_ZDSTSOCK in all uppercase.

■ Use the prefix **NVDARG** to override property settings.

For example, if you want to override the value set for the identification property, which identifies the subscriber session to the Radia Configuration Server, type:

```
Session.Property("NVDARGIDENTIFICATION")="jenns"
```

■ Use the prefix **NVDOBJ** to override object attributes.

For example, if you want to override the value set for the ZDSTSOCK attribute of the ZMASTER object, which is the port setting for the Radia Configuration Server, type:

```
Session.Property("NVDOBJZMASTER_ZDSTSOCK")="3462"
```

Using a Post-Install Script

Use custom Visual Basic, REXX, or Tcl scripts to run processes after installing Radia Clients. For example, your post-install script can initiate a connection to the Radia Configuration Server in order to process mandatory applications. A sample script is available in the **\win32\samples** folder on the Radia Client installation CD-ROM.

Caution

Be sure to use the NVDPOSTCAPATH argument to specify the fully qualified path and filename of the custom Visual Basic or REXX post-install script in INSTALL.INI or on the command line. See the description of NVDPOSTCAPATH in Table 2.2 ~ [PROPERTIES] Section of INSTALL.INI on page 37. For example, if you want to run a script called redstart.rex, uncomment and set NVDPOSTCAPATH=C:\Progra~1\Novadigm\radstart.rex

---

Include the script in the \maint folder of the client install. It will automatically get copied into IDMSYS. A script example is shown below:

```
/***************************************************************************/
/** RADSTART.REX                                                        **/
/**                                                                     **/
/** DESCRIPTION:                                                        **/
/** Client Rexx will perform a Radia connection to an RCS defined in the **/
/** install.ini to process all mandatory applications.                  **/
/**                                                                     **/
/** AUTHOR:           HP                                          **/
/** LANGUAGE:         REXX                                              **/
/**                                                                     **/
/***************************************************************************/
/* trace i */

fullcmd = 'HIDE radntfyc localhost wait radskman
context=m,log=connect_initial.log'
call edmcmd fullcmd;
```

Figure 2.26 ~ Sample REXX code from a post-install script.

Summary

■   The Radia Client installation program consists of one package with three feature sets: Radia Application Manager, Radia Inventory Manager, and Radia Software Manager.

■   You can install the Radia Clients using a command line or using the Installation Wizard.

■   Customize your installation to suit your enterprise.

■   Create pre-install and post-install scripts to run processes before or after the clients are installed.

3

Installing the Radia Administrator Workstation

At the end of this chapter, you will:

■ Understand the system requirements for installing the Radia Administrator Workstation.

■ Be familiar with the installation files.

■ Know how to install the Radia Administrator Workstation using the Installation Wizard and command lines.

■ Understand the feature settings of the Radia Administrator Workstation.

■ Know how to remove and repair the Radia Administrator Workstation using the Installation Wizard and command lines.

The Radia Administrator Workstation installation program uses Microsoft Windows Installer. The program consists of one MSI package with four feature sets—Radia Packager, Radia System Explorer, Radia Client Explorer, and Radia Screen Painter.

System Requirements

- Clean computer. (A *clean computer* is a computer with only the target subscriber's operating system installed.)
- Windows 95, 98, NT 4.0, 2000, or XP.
- TCP/IP connection to the Radia Configuration Server.
- Minimum resolution of 800 x 600.
- MS Windows Installer Version 2.0 or higher.
  The MSI 2.0 installation program is available in the **managementinfrastructure \administratorworkstation\win32\msi** folder on the Radia Infrastructure CD-ROM. If Windows Installer does not exist, or if an earlier version is detected on the computer, the MSI 2.0 installation program runs automatically.
- For Windows NT, 2000, Server 2003, or XP, you must have administrator rights to the computer to install the Radia Administrator Workstation.

About the Installation Files

setup.exe

setup.exe is stored on the Radia Infrastructure CD-ROM in the **managementinfrastructure\administratorworkstation\win32** folder. It accepts any command line parameters and passes them to Windows Installer.

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

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

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

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

The target directory contains RADADMIN40.MSI, the installation folders, and setup.exe.

RADADMIN40.MSI

RADADMIN40.MSI is the MSI database file, which contains the default configuration information for the installation. This file is stored on the Radia Infrastructure CD-ROM in the **managementinfrastructure\administratorworkstation\win32** folder.

Installing the Radia Administrator Workstation

This section describes how to install the Radia Administrator Workstation using the Installation Wizard and using a command line.

Using the Installation Wizard to Install the Radia Administrator Workstation

This section describes how to install the Radia Administrator Workstation for Windows using the Installation Wizard.

To install the Radia Administrator Workstation using the Installation Wizard

1. From the folder containing the Radia Administrator Workstation installation files, run **setup.exe**.

The **Radia Administrator Workstation Installation Wizard** opens.
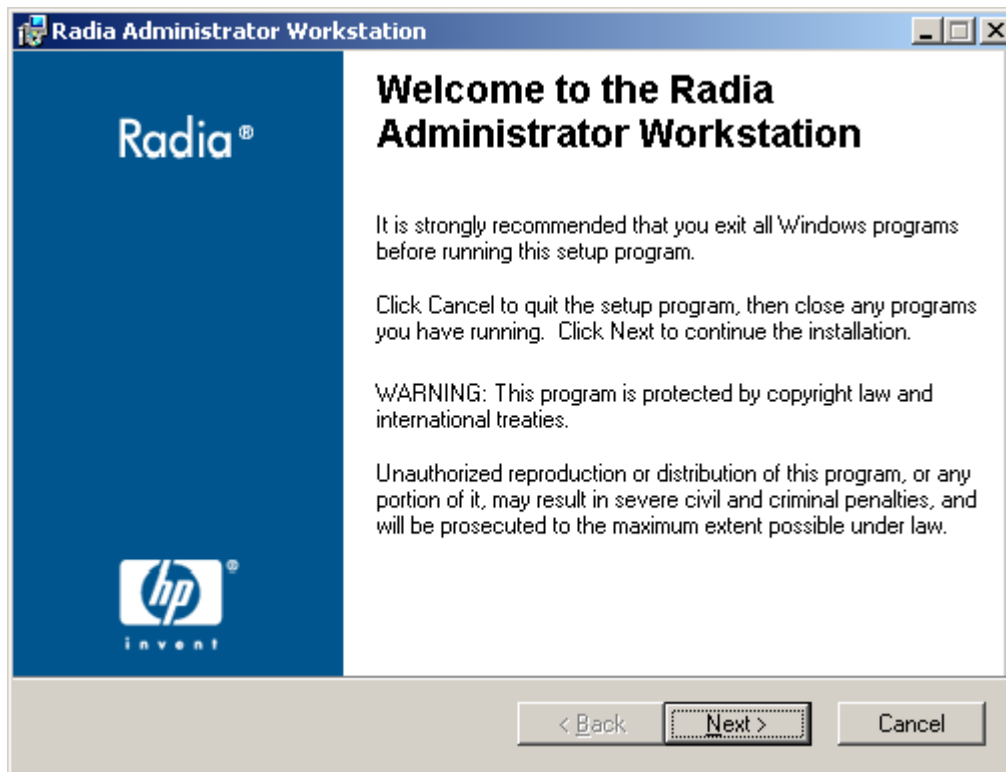
Figure 3.1 ~ Welcome window.

2. Click **Next**.

The **License Agreement** window opens.

Figure 3.2 ~ Read the License Agreement.

3. After reading and accepting the license agreement, click **Next**.

If the Radia Client is not installed on the computer, the **Destination Folder** window opens.

> **Note**
>
> If the Radia Client is already installed on the computer, this window will not open and the Radia Administrator Workstation is installed in the same location as the Radia Client.
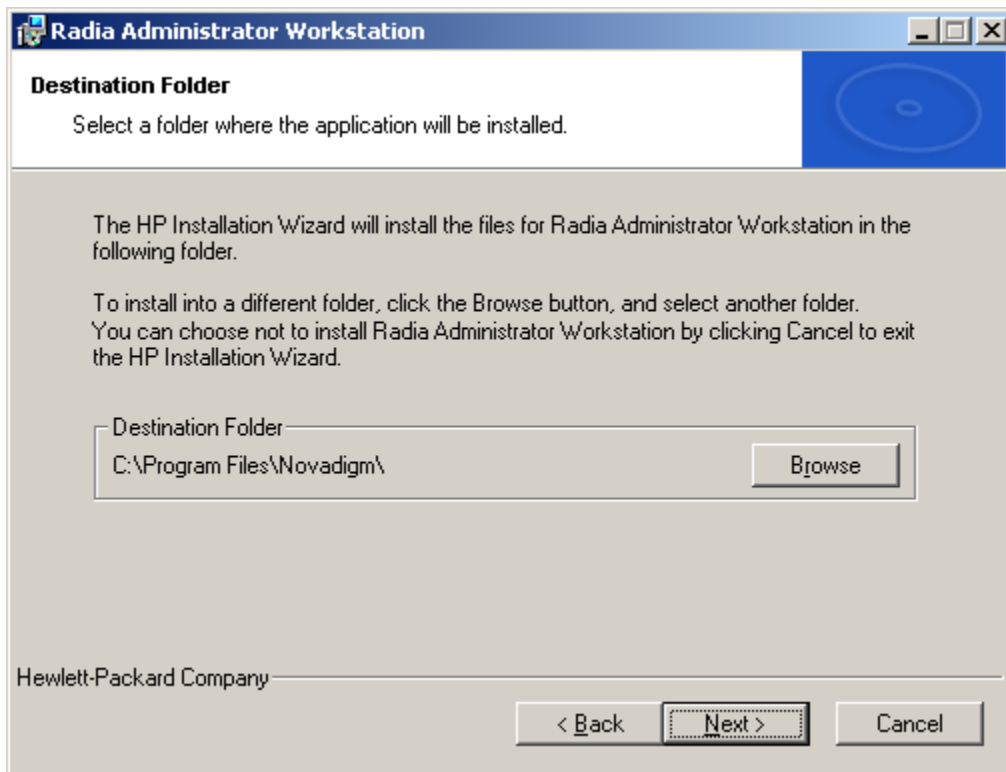
Figure 3.3 ~ Destination Folder window.

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

Click **OK** to continue.

4. Click **Next**.

The **Radia Configuration Server** window opens.
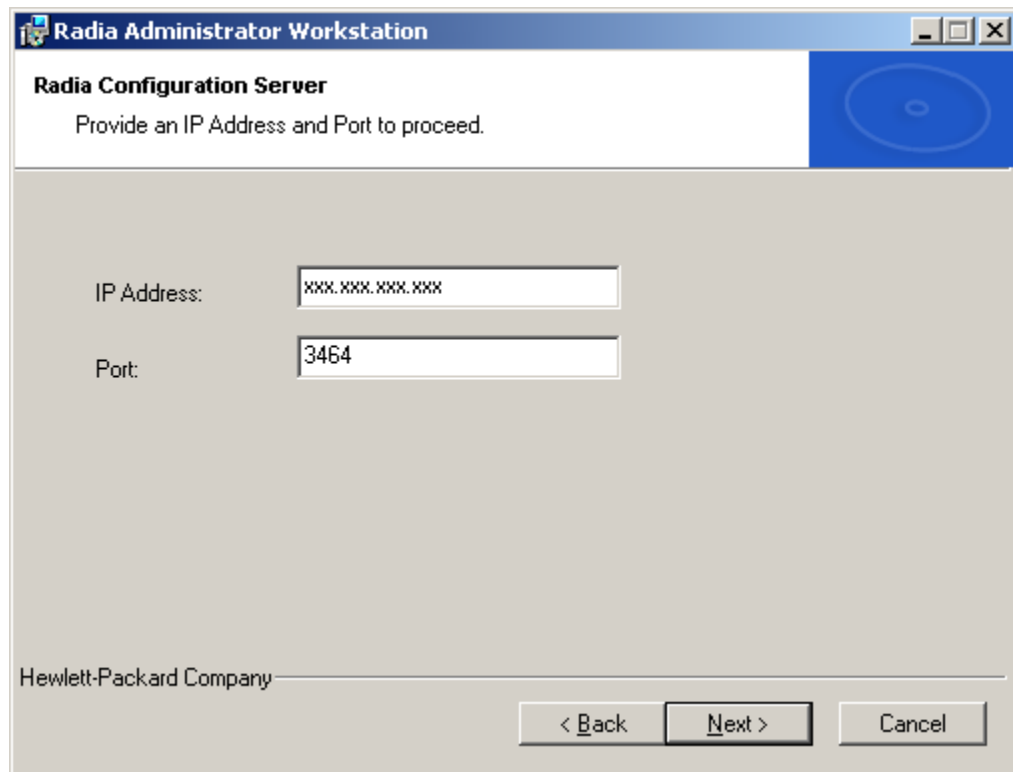
Figure 3.4 ~ Radia Configuration Server window.

5. In the **IP Address** text box, type the IP address for the Radia Configuration Server.
6. In the **Port** text box, type the port number (default is 3464).
7. Click **Next**.

The **Select Features** window opens.
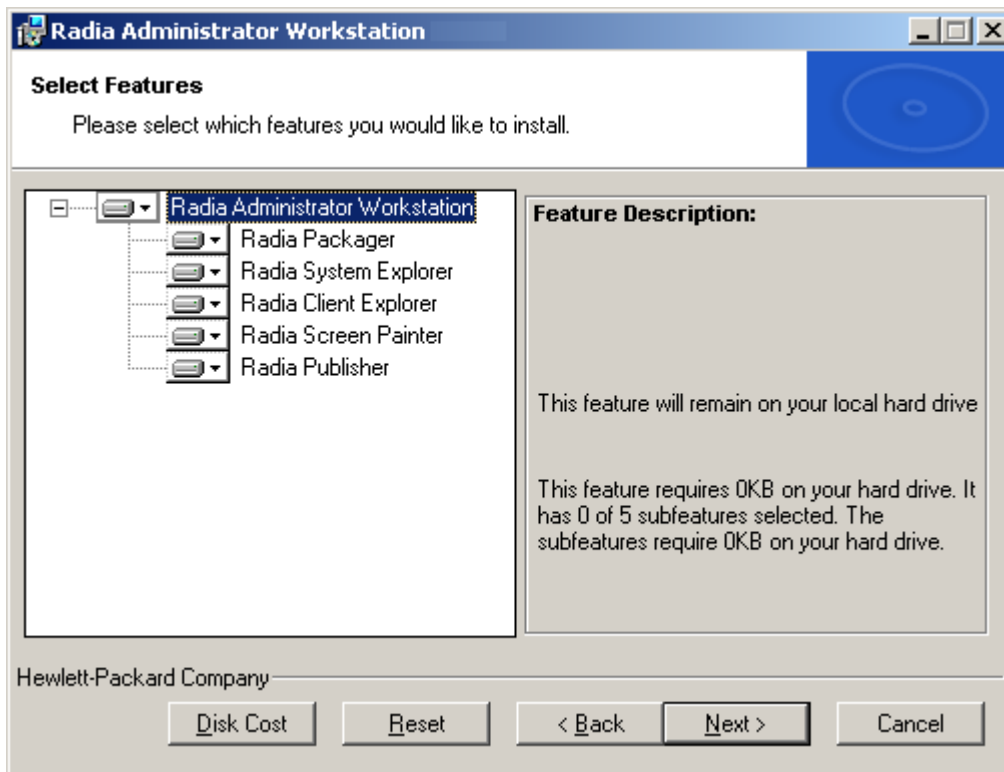
Figure 3.5 ~ Select Features window.

8. Click [icon] to select the features that you would like to install.

> **Tips**
>
> - If you want to set the same options for all of the features, you can click [icon] next to **Radia Administrator Workstation** and select the appropriate option to apply the setting to all features.
> - Click **Disk Cost** to see an overview of the disk space needed for the installation.

Each time you click [icon] a shortcut menu for that feature opens.
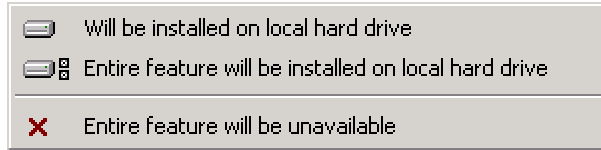
Figure 3.6 ~ Feature set installation options.

9. From the shortcut menu, select an installation option. These options are described in Table 3.1 below.

Table 3.1 ~ Feature Settings for the Radia Administrator Workstation	
Option	**Description**
Will be installed on local hard drive	Installs the top-level feature on the local hard drive, but not any sub-features listed below.
Entire feature will be installed on local hard drive	Installs the entire feature, including any sub-features listed below. **Note:** In this installation program, selecting this option or the "Will be installed on local hard drive" option for any of the features results in the same installation because these features do not contain sub-features.
Entire feature will be unavailable	The feature will not be installed. If previously installed, this feature will be removed.

10. Click **Next**.

The **Ready to Install the Application** window opens.
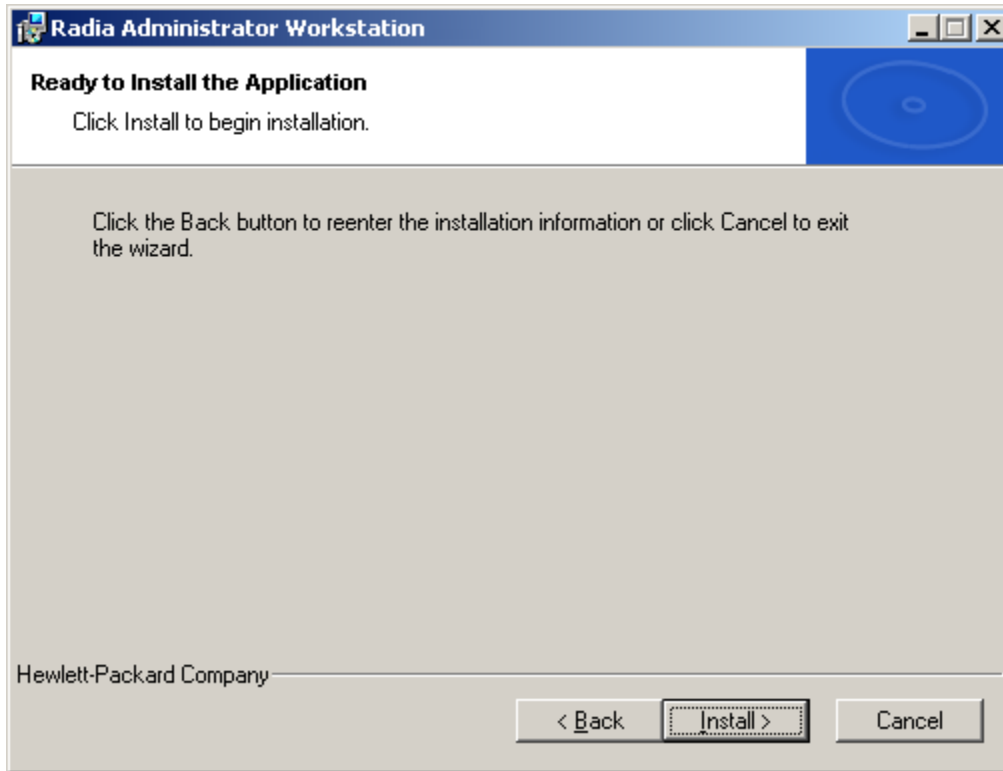
Figure 3.7 ~ Ready to Install the Application window.

11. Click **Install** to begin the installation.

When the installation is done, the **Radia Administrator Workstation 4.x has been successfully installed** window opens.
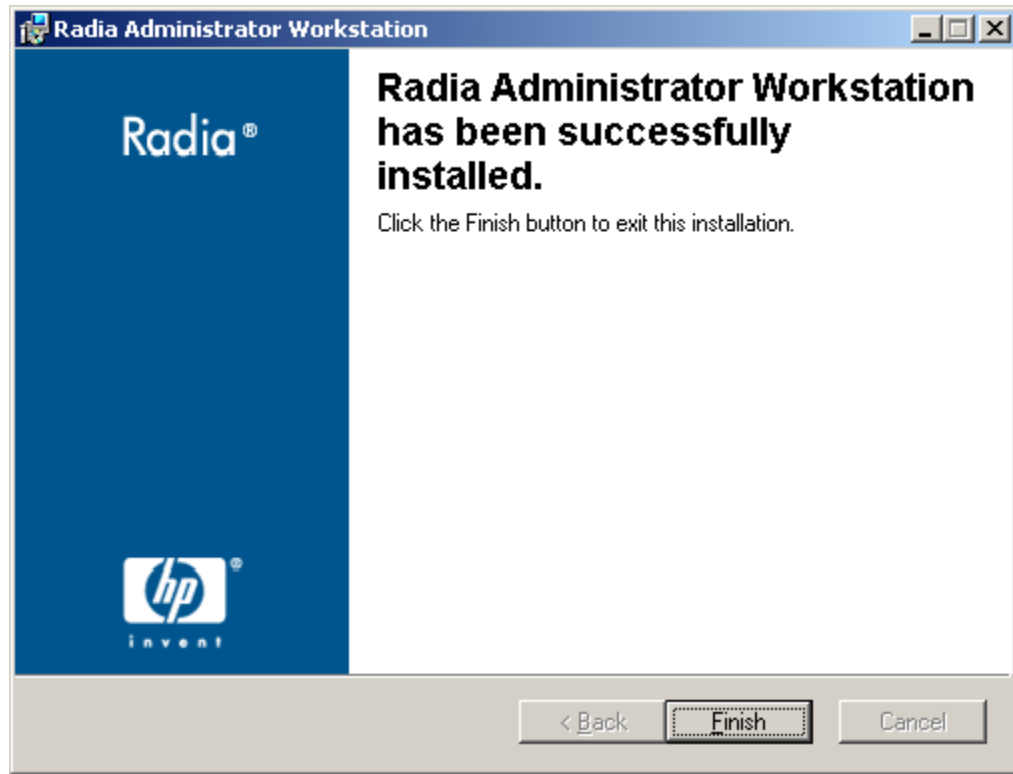
Figure 3.8 ~ Successful installation window.

12. Click **Finish** to exit the installation.

Using a Command Line to Install the Radia Administrator Workstation

You can also use the command line to run the Radia Administrator Workstation installation program For example, if you want to install only the Radia Packager on a computer, the command line that you run from the directory containing the Radia Administrator Workstation installation files might be:

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

Specifying the Features to Install

To specify the features that you want to install, use the appropriate feature state argument, such as ADDLOCAL, and specify the features that you want to install.

Table 3.2 ~ Radia Administrator Workstation Feature State Arguments	
Specify the following arguments	**To set the feature state**
ADDLOCAL	Type a comma-delimited list of features that you want set to "Will be installed on local hard drive."
ADDSOURCE	Type a comma-delimited list of features that you want set to "Will be installed to run from network."
ADVERTISE	Type a comma-delimited list of features that you want set to "Feature will be installed when required."
REMOVE	Type a comma-delimited list of features that you want set to "Entire feature will be unavailable."
	This only removes the features—not the entire product. Therefore, if you use the REMOVE property and type each of the feature names, the core product will still be stored on your computer.
	If you want to remove the entire product, type REMOVE=ALL.

When specifying features on the command line, reference the Radia Administrator Workstation features as follows:

- NVDINSTALLPACKAGER   Radia Packager
- NVDINSTALLPUBLISHER   Radia Publisher
- NVDINSTALLSYSTEMEXPLORER   Radia System Explorer
- NVDINSTALLCLIENTEXPLORER   Radia Client Explorer
- NVDINSTALLSCREENPAINTER   Radia Screen Painter

For example, if you want to install the Radia System Explorer and the Radia Client Explorer to the computer, the command line might be:

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

Caution
If you run the installation from a command line, be sure to pass the IP address for the Radia Configuration Server to the installation. For example: ```SETUP.EXE NVDOBJZMASTER_ZIPADDR=10.10.10.1```

Additional Command Line Arguments

Some additional arguments that you can pass to the installation program on the command line are described in Table 3.3 below.

Table 3.3 ~ Command Line Arguments	
Argument	**Description**
/qn	Performs a silent installation.
/qb	Displays the progress bar only during the installation.
/l*v *drive*:*install.log*	Creates a detailed Windows Installer log. **Note:** Using this option may impact the performance of the installation.
/a TARGETDIR= *drive*:*targetdirectory*	Creates a Windows Installer AIP in the specified target directory. **Note:** A Windows Installer Administrative Installation Point (AIP) is also known as an Administrative Control Point (ACP). The target directory contains RADADMIN40.MSI, the installation folders, and setup.exe. Once you have created the AIP, you can run setup.exe and pass the appropriate command line parameters. This starts the Windows Installer and passes the specified parameters to it.

Removing the Radia Administrator Workstation

The Windows Installer installation program offers the ability to remove the Radia 4.x Administrator. This section describes how to remove the Radia Administrator Workstation using the Installation Wizard and using a command line.

Using the Installation Wizard to Remove the Radia Administrator Workstation

This section describes how to remove (uninstall) the Radia Administrator Workstation using the Installation Wizard.

Note

To remove specific features of the Radia Administrator Workstation, use the Modify option on the Application Maintenance window. This is discussed in *Modifying the Radia Administrator Workstation Installation* on page 107.

To remove the Radia Administrator Workstation using the Installation Wizard

1. From the folder containing the Radia Administrator Workstation installation files, double-click **setup.exe**.

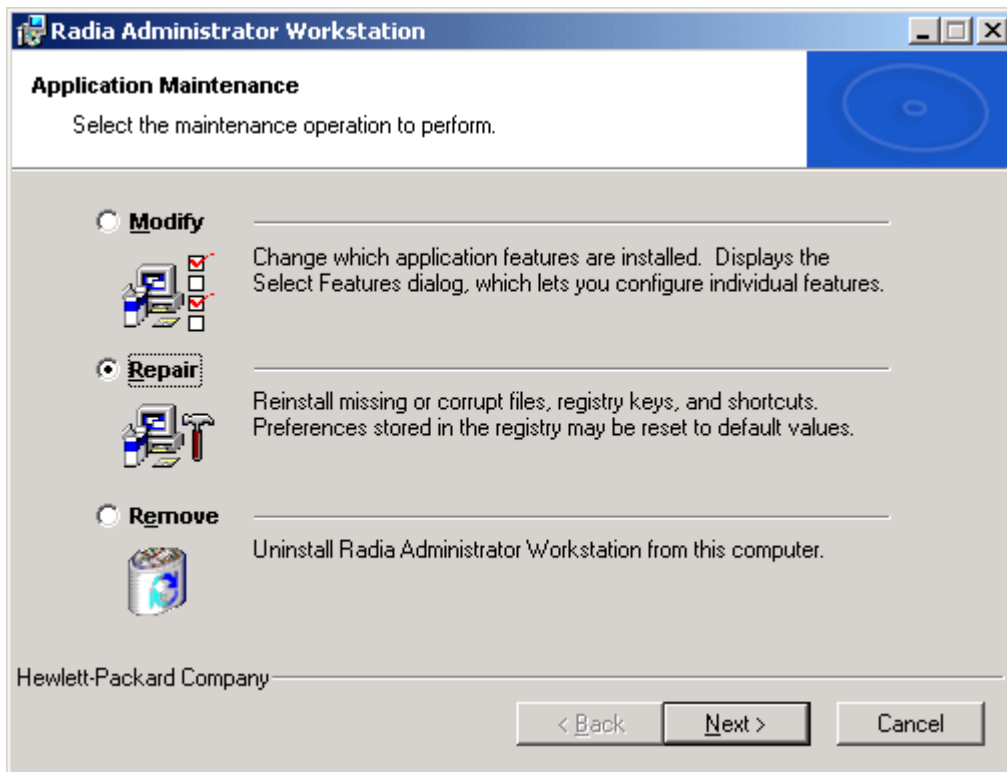The **Application Maintenance** window opens.
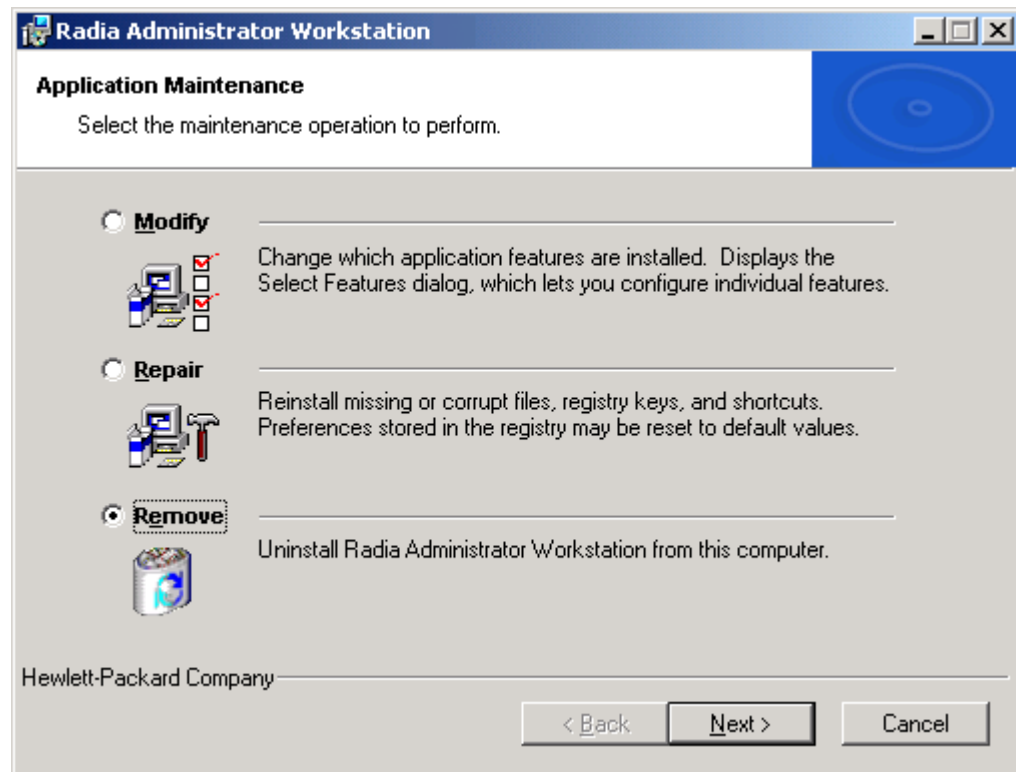
Figure 3.9 ~ Application Maintenance window.

2. Select the **Remove** option.
3. Click **Next**.

The **Radia Administrator Workstation 4.x Uninstall** window opens.
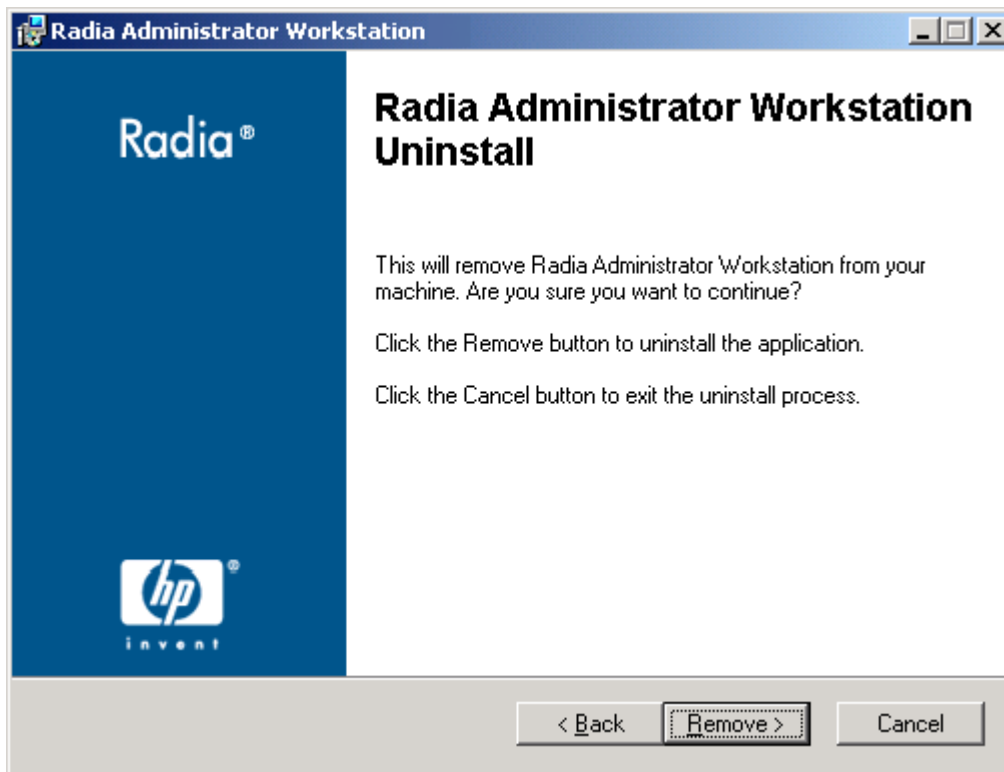
Figure 3.10 ~ Radia Administrator Workstation Uninstall window.

4. Click **Next**.

The files for the Radia Administrator Workstation are removed from the computer.

The **Radia Administrator Workstation has been successfully uninstalled** window opens.
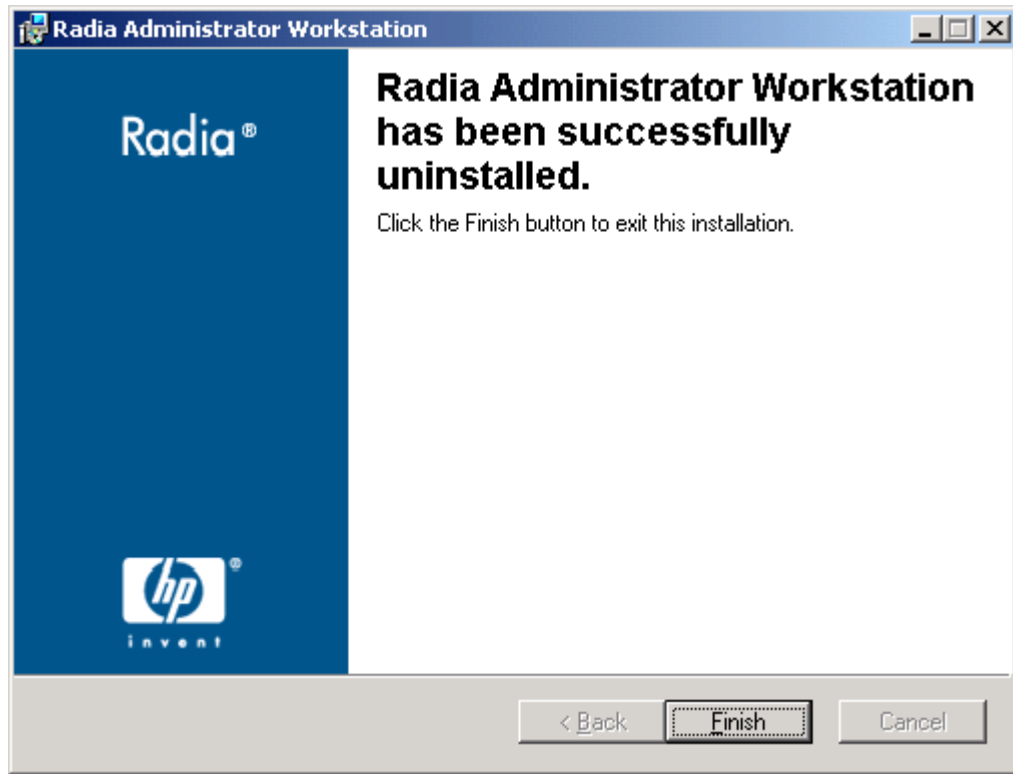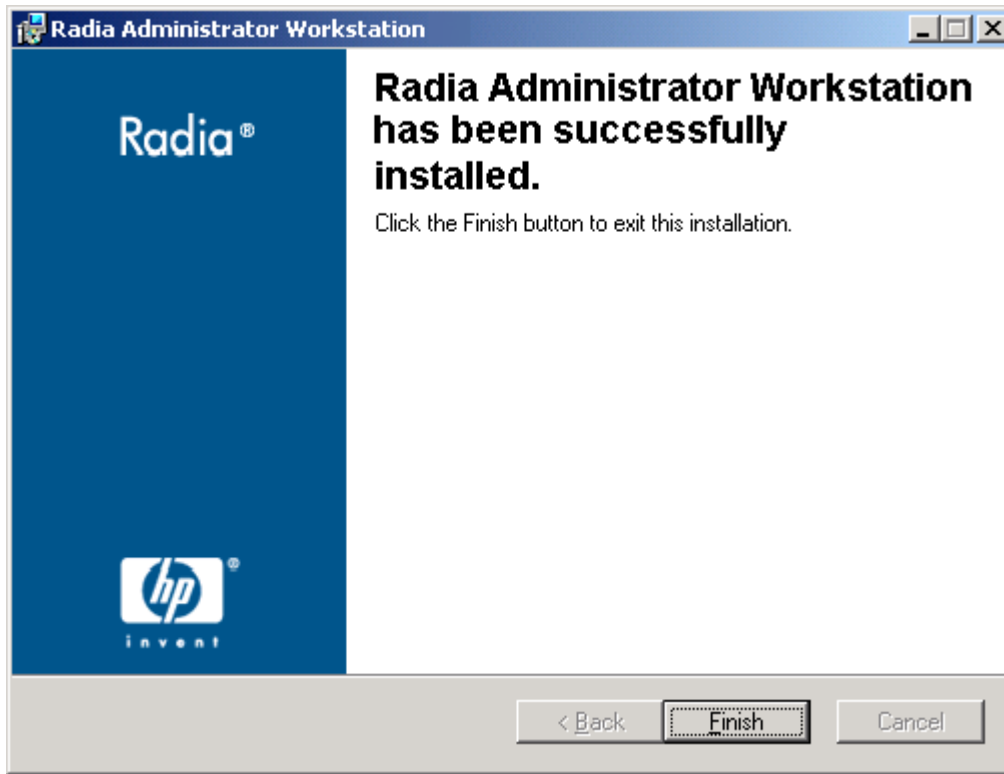

Figure 3.11 ~ Successful Uninstallation window.

5. Click **Finish**.

Using a Command Line to Remove the Radia Administrator Workstation

This section describes how to remove (uninstall) the Radia Administrator Workstation using a command line.

To remove the Radia Administrator Workstation using a command line

■ From the folder containing the Radia Administrator Workstation installation files, type the following command line:

```
SETUP.EXE REMOVE=ALL
```

OR

If you would like to remove a single Radia Administrator Workstation feature, type a comma-delimited list of the features that you want to remove on the command line.

EXAMPLE

If you want to silently remove the Radia System Explorer and Radia Client Explorer, type:

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

Note
Reference the features for the Radia 4.x Administrator Workstations as follows: · Radia Packager NVDINSTALLPACKAGER · Radia Publisher NVDINSTALLPUBLISHER · Radia System Explorer NVDINSTALLSYSTEMEXPLORER · Radia Client Explorer NVDINSTALLCLIENTEXPLORER · Radia Screen Painter NVDINSTALLSCREENPAINTER

Caution
This only removes the features—not the entire product. Therefore, if you use the REMOVE property and type each of the feature names, the core product will still be stored on your computer.

Repairing the Radia Administrator Workstation

The Windows Installer installation program offers the ability to repair the Radia Administrator Workstation. For example, if you have a missing Radia Administrator Workstation module, you can use this tool to repair the installation. This tool will not overwrite modules that exist on the computer if they are newer than the ones provided with the installation.

This section describes how to repair the Radia Administrator Workstation using the Installation Wizard and using a command line.

Using the Installation Wizard to Repair the Radia Administrator Workstation

This section describes how to repair the Radia Administrator Workstation using the Installation Wizard.

To repair the Radia Administrator Workstation using the Installation Wizard

1. From the folder containing the Radia Administrator Workstation installation files, double-click **setup.exe**.

   The **Application Maintenance** window opens.

Figure 3.12 ~ Application Maintenance window.

2. Select the **Repair** option.
3. Click **Next**.

The **Ready to Repair the Application** window opens.
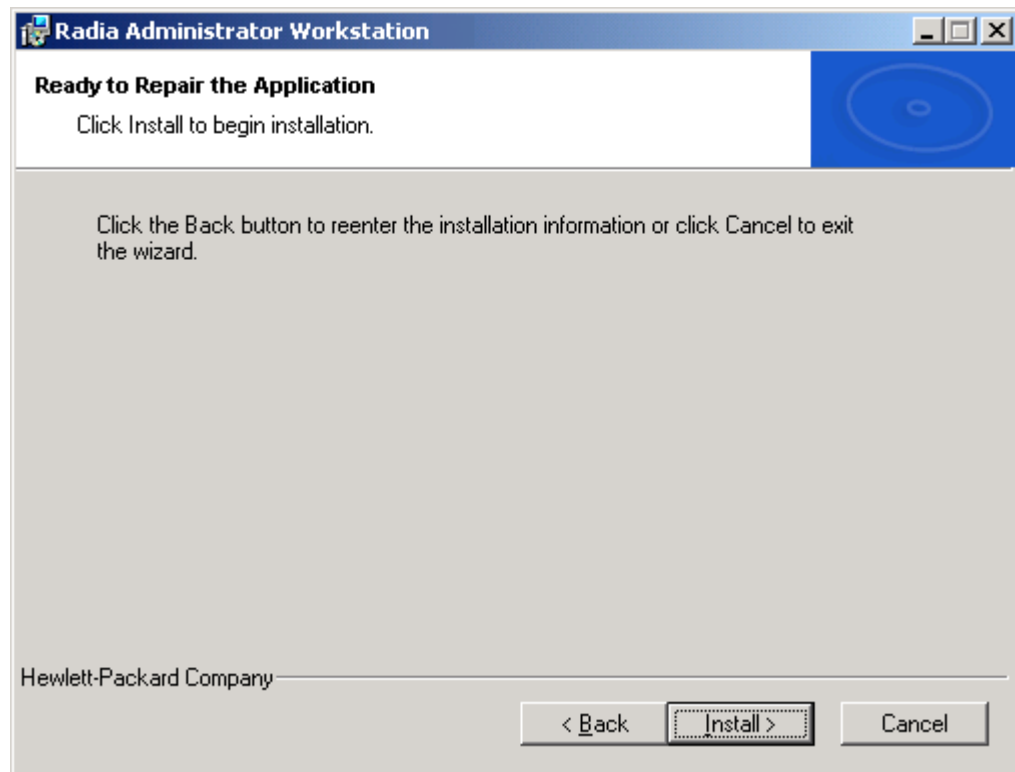
Figure 3.13 ~ Ready to Repair the Application window.

4. Click **Next**.

When the repair is done, the **Radia Administrator Workstation has been successfully installed** window opens.

Figure 3.14 ~ Successful installation window.

5. Click **Finish**.

Using a Command Line to Repair the Radia Administrator Workstation

This section describes how to repair the Radia Administrator Workstation using a command line.

To repair the Radia Administrator Workstation using a command line

■   From the folder containing the Radia Administrator Workstation installation files, type the following command line:

```
msiexec /f radadmin40.msi
```

Note
Additional parameters can be used with this command line. For more information, see your Windows Installer documentation.

Modifying the Radia Administrator Workstation Installation

The Windows Installer installation program offers the ability to modify the Radia 4.x Administrator Workstation installation by adding or removing individual features. This section describes how to modify the installation of the Radia Administrator Workstation using the Installation Wizard and using a command line.

Using the Installation Wizard to Modify the Radia Administrator Workstation

This section describes how to modify the installation of the Radia Administrator Workstation using the Installation Wizard.

To modify the Radia Administrator Workstation installation using the Installation Wizard

1. From the folder containing the Radia Administrator Workstation installation files, double-click **setup.exe**.

The **Application Maintenance** window opens.
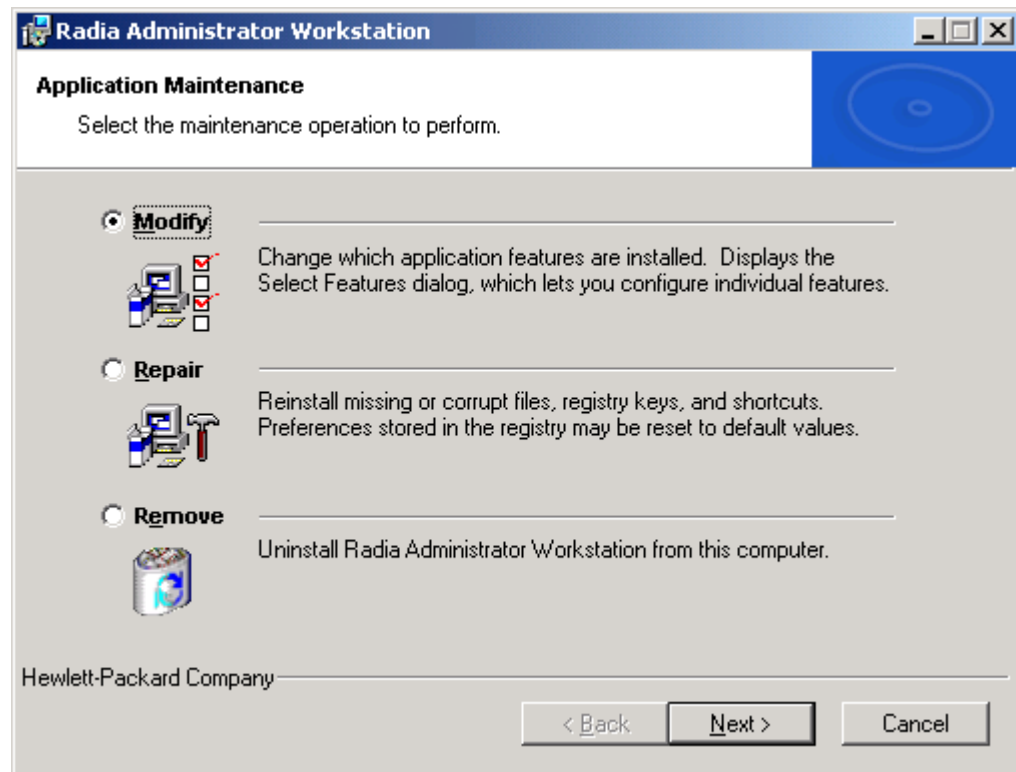
Figure 3.15 ~ Application Maintenance window.

2. Select the **Modify** option.

3. Click **Next**.

The **Select Features** window opens. See *Installing the Radia Administrator Workstation* on page 86 for information about how to use this window.
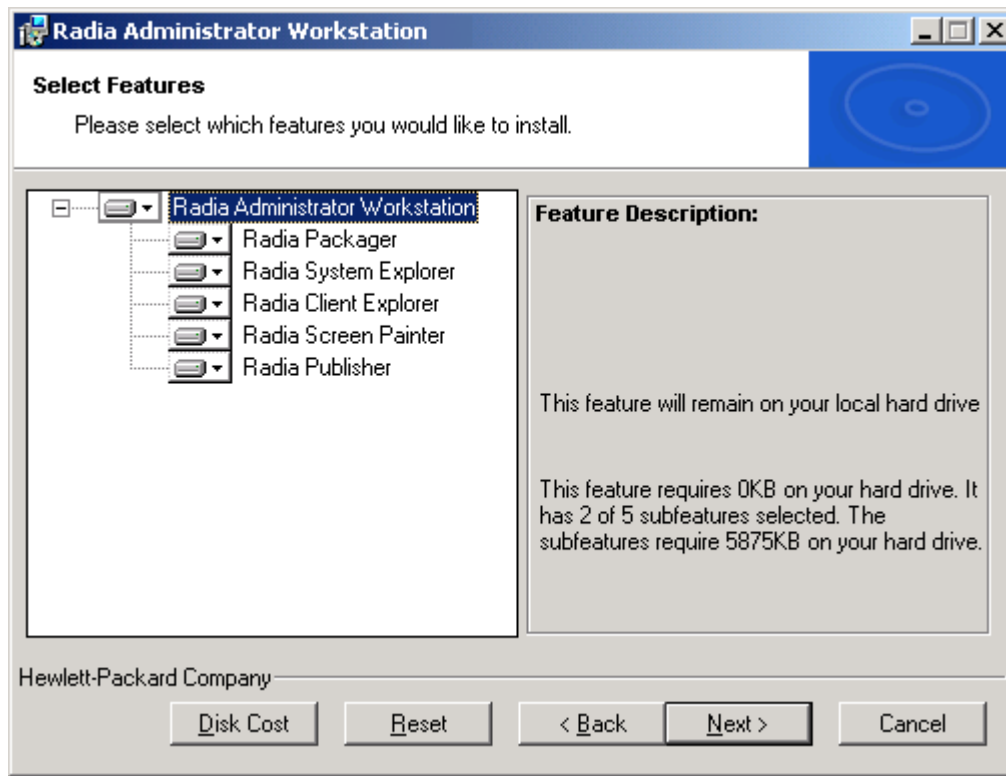
Figure 3.16 ~ Select Features window.

4. Click **Next**.

The **Ready to Modify the Application** window opens.
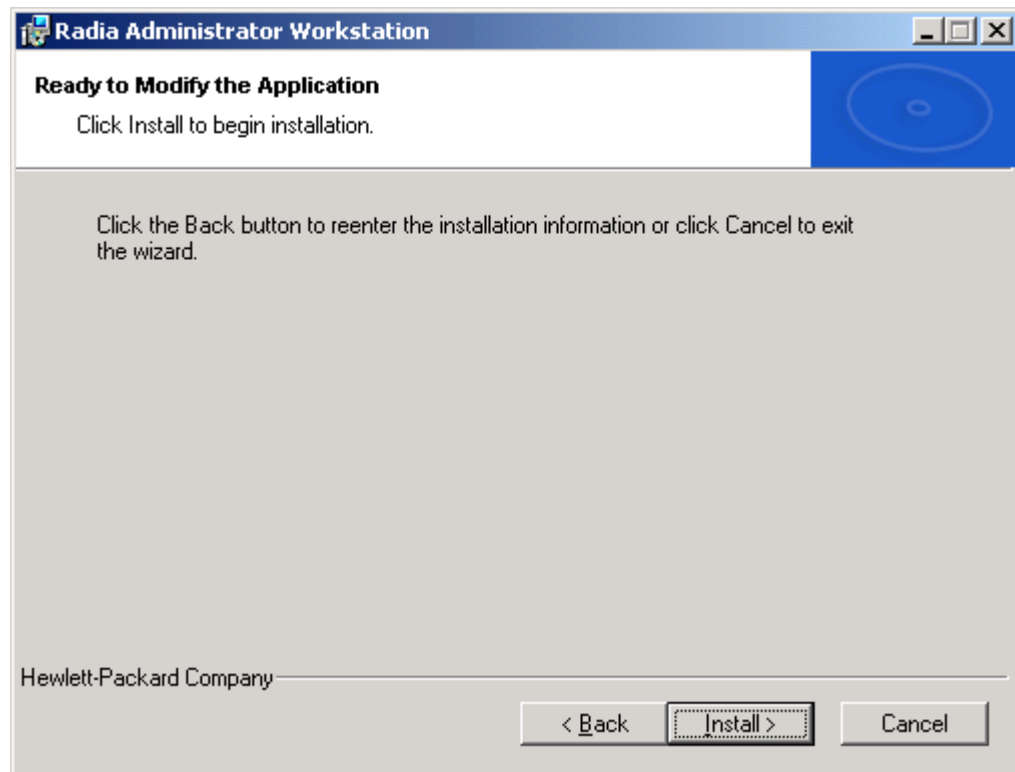
Figure 3.17 ~ Ready to Modify the Application window.

5. Click **Next**.

The **Radia Administrator Workstation 4.x has been successfully installed** window opens.
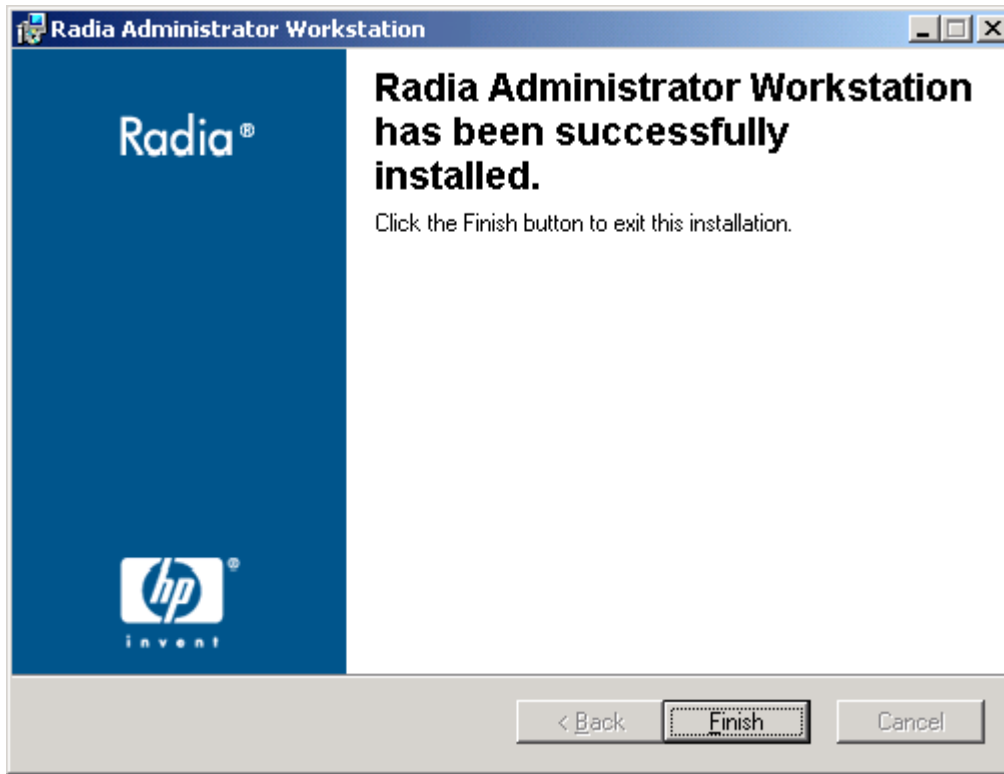
Figure 3.18 ~ Successful installation window.

6. Click **Finish** to close the installation program.

Using a Command Line to Modify the Radia Administrator Workstation Installation

To modify the Radia Administrator Workstation installation using a command line

■ From the folder containing the Radia Administrator Workstation installation files, type the following command line:

```
SETUP.EXE FeatureStateArgument=feature1,feature2
```

See Table 3.2 on page 95 for more information.

EXAMPLE

If you want to install the Radia Packager to the local hard drive, and to make the Radia System Explorer and Radia Client Explorer unavailable, use the following command line:

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

See *Additional Command Line Arguments* on page 96 for additional arguments.

Summary

■ The Radia Administrator Workstation consists of one package with four feature sets: Radia Publisher, Radia System Explorer, Radia Client Explorer, and Radia Screen Painter.

■ Install the Radia Administrator Workstation on a clean computer.

■ You can install the Radia Administrator Workstation using a command line or using the Installation Wizard.

4

Packaging Applications and Content

At the end of this chapter, you will:

■ Understand the packaging process.

■ Be able to package an application using Component Selection Mode or Installation Monitor Mode.

■ Be aware of the Radia Publishing Adapter, Radia Extensions for Windows Installer, and the Radia Publisher.

■ Use the New Application Wizard in the Radia System Explorer to create a service.

This guide covers the *suggested* implementation for the Radia Software Manager. Although you will tailor this strategy to meet your organization's needs, we recommend that you review this guide for a comprehensive understanding of the Radia Software Manager. This chapter focuses on packaging.
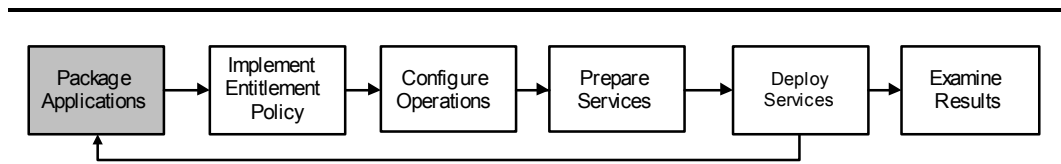
Figure 4.1 ~ Overview of the Radia Software Manager Guide.

Note
The Radia Publisher for Radia Version 3.x has been renamed the Radia Packager for Radia Version 4.x. The Radia Packager is similar to the Radia 3.x Publisher except that it only packages in component selection mode and installation monitor mode. Packaging for applications installed or maintained using Windows Installer is now handled by the Radia Publisher. The Radia Publisher is covered in the Radia Publisher Guide.

About Packaging and Publishing with Radia

Packaging is the process of identifying resources, editing those resource's installation attributes, defining how they are to be installed, and saving the resources and installation instructions in a machine-readable file format. A package typically contains one or more files and configuration settings.

Publishing is the process of importing a package and its imbedded information to the Radia Database (AKA the Radia Configuration Server Database). A package must be published before its content can be distributed and deployed into your environment.

Radia offers many different packaging and publishing options through multiple tools. Make note of the different types of resource packages you will create and publish, then choose your Radia packaging and publishing products as required.

■   If you're **packaging** Windows Installer files, use the **Radia Packager for Windows Installer**. (See the *Radia Extensions for Windows Installer Guide*).

■   If you're **publishing** Windows Installer files, use the **Radia Publisher**. (See the *Radia Publisher Guide).*

■   To **package** and **publish** all other types of files and applications, use the **Radia Packager**. (See the *Radia Application Manager Guide* or the *Radia Software Manager Guide*).
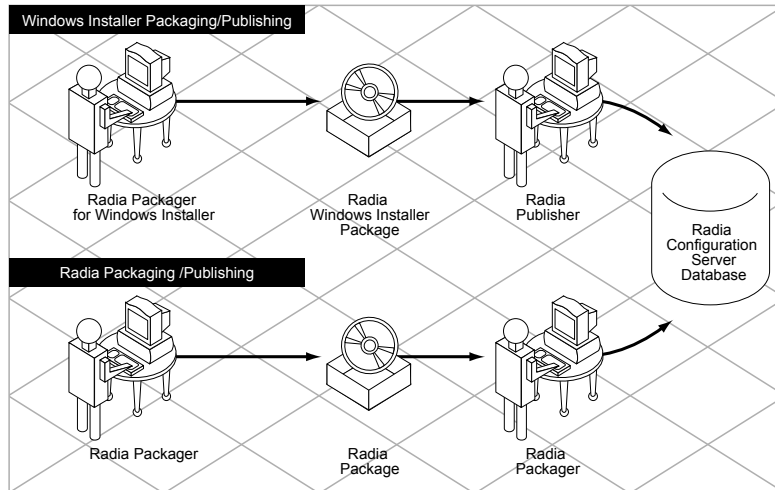
Figure 4.2 ~ Radia Packaging/Publishing options.

About Radia Packager

Packaging is the process of identifying the components of the software and organizing them into *packages*. Packages contain the files, shortcuts, links, and/or registry entries that make up the software. The software that you distribute can vary greatly—from a single data file, such as a company telephone list, to an entire application suite, such as Microsoft Office 2000.

Two packaging modes are available in the Radia Packager:

■ **Component Selection Mode**
In Component Selection Mode, you select the individual components that make up the application, such as files, directories, registry entries, and links. We recommend this mode for packaging simple data or applications.

■ **Installation Monitor Mode**
In Installation Monitor Mode, the Radia Packager determines what to package by scanning the computer before and after installing the software. It *differences* the before and after scans to determine what changes were made to the computer. These differences make up the package that you promote to the Radia Database. We recommend this mode for packaging when you do not know all of the components that make up the application.

> **Note**
>
> The term *computer* is used to refer to a workstation or server.

Several factors influence which mode you will use, the complexity of the application, your knowledge of the application's structure, and whether you would like to use the application's native installation capabilities. Later in this chapter, you will find detailed coverage of these modes.

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

- An **Application Packages (PACKAGE)** instance that represents the promoted package.

- One **File Resources (FILE)** instance for each file in the package.

- One **Desktop (DESKTOP)** instance for each program group, link, and shortcut in the package.

- One **Path (PATH)** instance for each unique path to one or more components on the computer where the software is installed.

- One **Registry Resources (REGISTRY)** instance for each hive in the package.

> **Note**
>
> Each instance described above is stored in one of the default classes in the SOFTWARE domain, such as the PACKAGE or FILE class. You can also add your own classes, such as a DLL class, to the Radia Database. See the *Radia System Explorer Guide* for information on how to add a class.
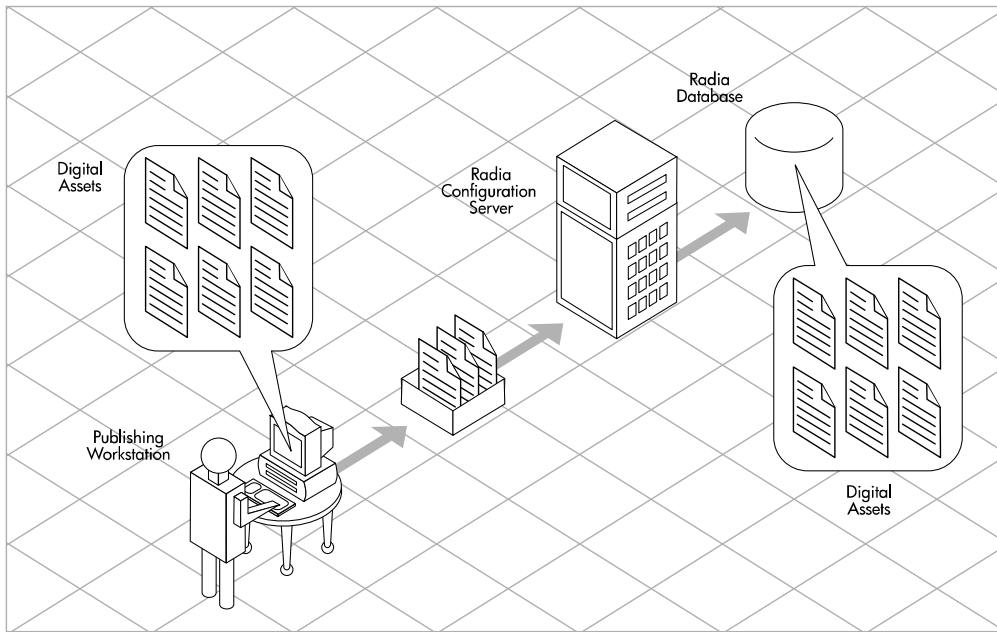
Figure 4.3 ~ Packaging digital assets.

After promoting the package, you will use the Radia System Explorer to create a service and assign policies. See *Chapter 5: Implementing Entitlement Policy* starting on page 233 for more information.

Packaging Considerations Checklist

Before packaging your data, there are several items that you need to consider.

General

❑   What do you know about the data to be packaged?

- ■   If it is a simple package and you know everything about it, use Component Selection Mode.

- ■   If it is a more complex package and you do not know everything about it, use Installation Monitor Mode.

❑   What is the name of the package going to be? Follow your naming conventions.

❑   Do you have a unique session ID? Follow your conventions.

System Configuration

See *Step 4: Setting the Required System Configuration* for Installation Monitor Mode on page 131, or for Component Selection Mode on page 179 for more information.

❑   What operating systems are your target computers (workstations or servers) using?

❑   How much RAM is necessary to handle the data?

❑   What is the minimum processor necessary to handle the data?

Activation Options

❑   When do you want to activate the application - immediately on distribution or later?

See *Step 5: Setting Date and Time Constraints* for Installation Monitor Mode on page 133, or for Component Selection Mode on page 181 for more information.

❑   Which version of the application do you want to distribute, and when do you want to activate it?

See the *Preparing Services* chapter starting on page 299 for more information.

Data Options

See *Data Options Tab* for Setting Default Packaging Properties on page 225, for Installation Monitor Mode on page 155, or for Component Selection Mode on page 194 for more information.

❑   What type of compression do you want to use?

❑   Will your data be stored in the Radia Database or on a Radia Proxy Server?

❑   Do you want to preserve the security settings for your files and folders that are being deployed to NTFS volumes?

❑   Are you distributing maintenance to the Radia Software Manager client?

❑   How do you want to promote the resources? Will you use the force lock method?

❑   Are you sending out an update and only want to deploy the changes?

❑   Do you want to enforce System File Protection (SFP) for Windows 2000 files and folders?

Verify Options

See *Client Management Tab* for Setting Default Packaging Properties on page 222, for Installation Monitor Mode on page 151, or for Component Selection Mode on page 190 for more information.

❑   Do you want to use the standard, default verification options?

❑   Is this a first time installation? Is there anything that you need to verify?

❑   When deploying files, what types of statistics do you want to check – date, time, size?

❑   Do you want to update all files, or only newer files?

❑   If a file already exists, do you want to deploy it again to overwrite any changes that may have been made?

❑   Do you want to use the internal version to determine whether a file should be deployed?

Delivery Options

See *Client Management Tab* for Setting Default Packaging Properties on page 222, for Installation Monitor Mode on page 151, or for Component Selection Mode on page 190 for more information.

❑   Do your files or methods need to be deployed in a particular order?

❑   Is the data mandatory or optional?

>   **Note:** You can only deliver mandatory files if the Radia Application Manager is installed.

❑   Do you want the data deployed under the user or machine context?

Client Behaviors

See *Client Behaviors Tab* for Setting Default Packaging Properties on page 228, for Installation Monitor Mode on page 158, or for Component Selection Mode on page 197 for more information.

❑   After the file is deployed, do you want to run any methods? If so, what are they?

❑   Does anything need to happen to enable the file once it's deployed? If so, what method will you run to enable it?

❑   If the subscriber is no longer subscribed to the software, do you want to delete the file?

❑   Do you want to compare the old and new version of the file that you are deploying? If so, what method do you want to use?

Using Installation Monitor Mode

In Installation Monitor Mode, the Radia Packager determines what to package by scanning the computer before and after installing the software. It *differences* the before and after scans to determine what changes were made to the computer after the installation. These differences make up the package that you will promote to the Radia Database.

We recommend this mode for packaging when you do not know all of the components that make up an application.

Packaging in Installation Monitor Mode involves five phases:

1. **Defining** the application's hardware, software, and operating system requirements.

2. **Analyzing** (scanning) the hardware and software on the computer.

3. **Installing** the software you are packaging.

4. **Analyzing** (scanning) the hardware and software on the computer again. The difference between the pre- and post-scans represents the software you installed.

5. **Promoting** the package to the Radia Database.

Recommendations

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

Before packaging your application in Installation Monitor Mode:

■ Set up a *clean computer*.

A clean computer has the same operating system as your target audience and the Radia Administrator Workstation installed. Do not make any other changes to the computer.

■ Use the same operating system and language version as the target computers for all packaging activities.

Caution

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

For example, if you are installing the French version of Office 2000 for Windows on computers using the French language version of Windows 9x, Windows NT, and/or Windows 2000 or Windows Server 2003, package the application on a computer with the French language version of the target operating system.

■ Package on a computer with a static IP address.

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

■ If you are packaging Visual Basic applications, create a Radia package that includes the Visual Basic 5 support files.

The Radia System Explorer, installed with the Radia Administrator Workstation, is a Visual Basic 5 application. Therefore, the following Visual Basic 5 support files are installed:

- ASYCFILT.DLL
- COMCTL32.OCX
- MSVBVM50.DLL
- OLEPRO32.DLL

- COMCAT.DLL
- COMDLG32.OCX
- OLEAUT32.DLL
- STDOLE2.TLB

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

To distribute the necessary support files with your Visual Basic applications, use Component Selection Mode to create a package that includes only the Visual Basic 5 support files. Then, include this package in any application package for Visual Basic 5. For more information on including packages, see *Inter-Package Relationships* in the *Packaging Strategy for the Radia Administrator* topic in the Radia Packager help.

The alternative is to package Visual Basic 5 applications on a computer that does not have the Radia System Explorer installed. To install the Radia Administrator Workstation without the Radia System Explorer, put the CD-ROM in the drive and type the following on the command line:

```
CDDrive:\radadmin.exe –ne
```

Packaging with Installation Monitor Mode

This section guides you through packaging a package using Installation Monitor Mode, and also provides detailed information about each of the screens that you will encounter.

In this example, you will package an evaluation copy of WinZip 8.0, a simple shareware application available at http://www.winzip.com. The target platforms are Windows 95, 98, NT, 2000, and Windows Server 2003.

Use this example to familiarize yourself with Installation Monitor Mode. However, please remember that there are many variables when packaging software.

Step 1: Logging On to Radia Packager

1. Go to **Start**, **Programs**, **Radia Administrator**, and select **Radia Packager**.

2. In the **Radia Packager Security Information** dialog box, type your **User ID** and **Password** in the appropriate text boxes.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

3. Click **OK**.

Step 2: Completing the Open Packaging Session Window

Use the **Open Packaging Session** window to select the packaging mode and enter information to identify the packaging session.
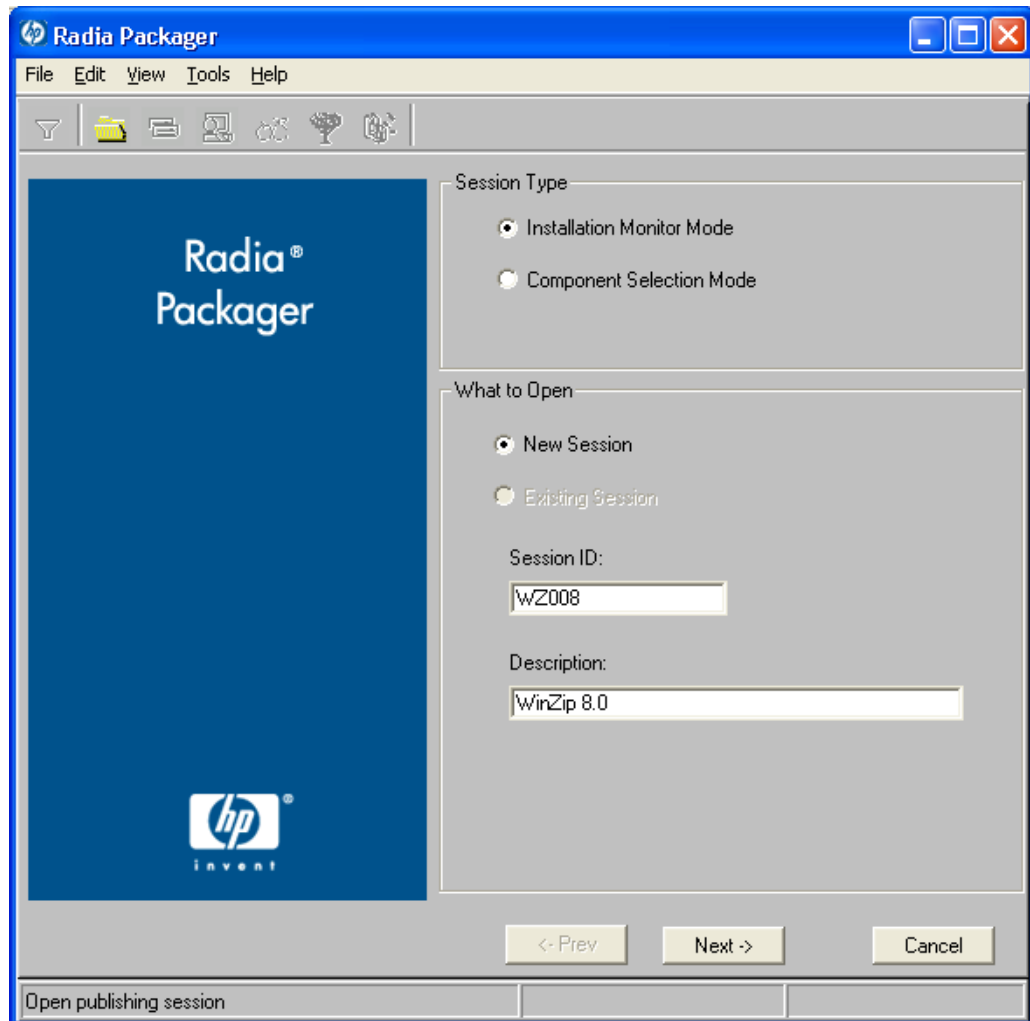
Figure 4.4 ~ Open Packaging Session window (Installation Monitor Mode).

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

The following sections describe the options on the **Open Packaging Session** window, as shown in Figure 4.4 on page 126.

Session Type Area

■ **Installation Monitor Mode**
Use this mode to create Radia packages when you do not know the structure of the software or data. Radia Packager performs a scan of your computer, and then prompts you to install the software using its native installation program. After the software is installed, Radia Packager performs another scan and then compares the scans. The differences are packaged and promoted to the Radia Database.

■ **Component Selection Mode**
Use this mode to create Radia packages when you know which files need to be distributed and what impact the installation will have on the client computer. In this mode, you identify the files that are packaged and then promoted to the Radia Database. For more information, see *Using Component Selection Mode* on page 171.

What to Open Area

■ **New Session**
Select **New Session** to begin a new packaging session. You must complete the **Session ID** and **Description** text boxes.

■ **Existing Session**
Select **Existing Session** to resume the previous session.

■ **Session ID**
Type a *unique* identifier that is one to six characters long.

■ **Description**
Type a description of the session.

In the WinZip example, as shown in Figure 4.4 on page 126, we are using **Installation Monitor Mode** to begin a **New Session**. The **Session ID** is **WZ008** and we've described the session as **WinZip 8.0**.

Click **Next** to go to the **Package Properties** window.

Step 3: Entering Package Properties

Use the **Package Properties** window to name the package and include additional descriptive information.
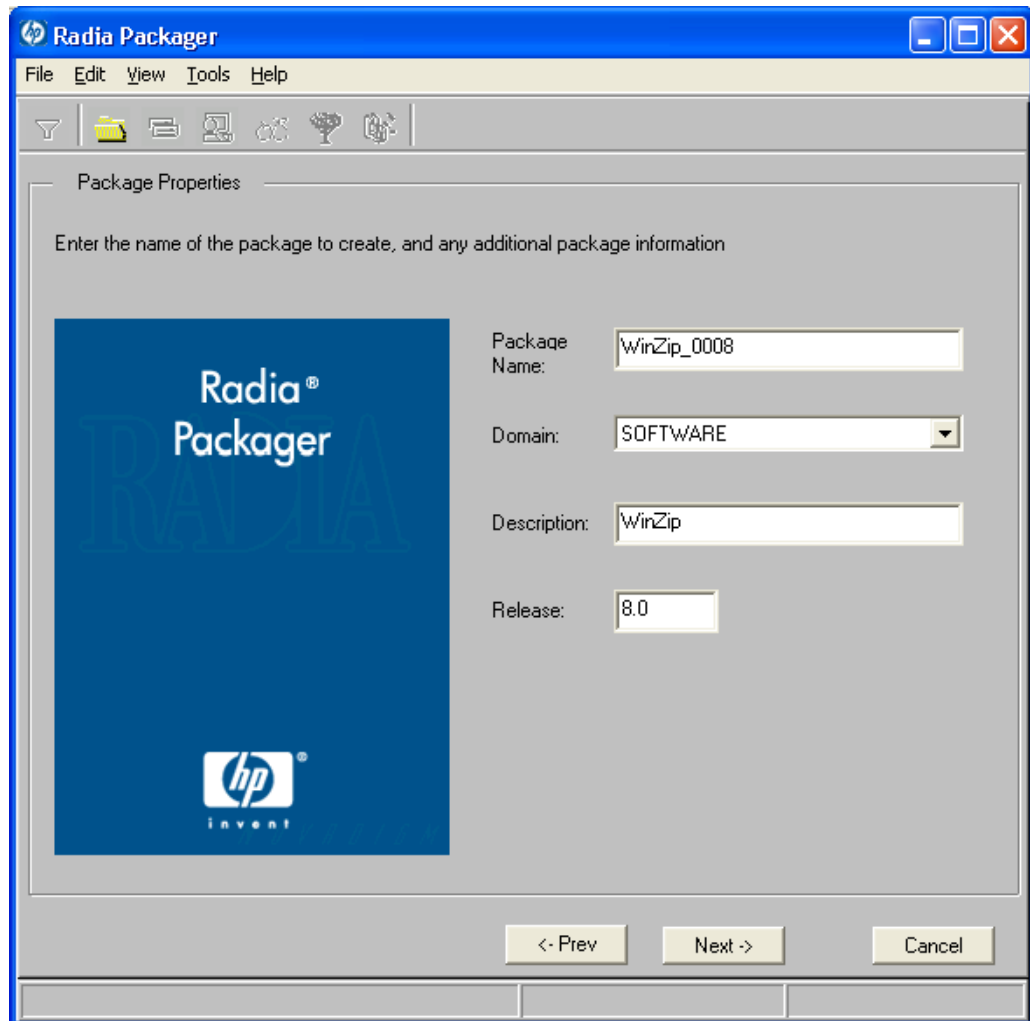
Figure 4.5 ~ Package Properties window (Installation Monitor Mode).

■ **Package Name**
Type a name for the package. This is the name for the PACKAGE class instance in the Radia

Database, and should conform to your naming conventions. Note that the name cannot contain any spaces.

> **Note**
>
> You may want to establish a naming convention to ensure that identifiers are unique. Radia Packager uses this identifier to construct data objects and filenames.
>
> See *Appendix A: Naming Conventions* starting 437 for more information.

- ■ **Domain**
  Select the domain to store the instance. This is normally the SOFTWARE domain unless you have customized the Radia Database with proprietary domains.

- ■ **Description**
  Type a description for the package.

- ■ **Release**
  Type the release number of the software.

In the WinZip example, as shown in Figure 4.5 on page 128, we named the package **WinZip_0008** and gave it a description and release number, **WinZip** and **8.0,** respectively.

> **Note**
>
> The current session information is stored *in* **SystemDrive:\Program Files\Novadigm \LIB\ZPAKSESS.EDM** on your computer. Use the Radia Client Explorer to view this object.
>
> Figure 4.6 on page 130 contains an example of ZPAKSESS.EDM.
>
> If you decide to close this packaging session and do not plan to complete the session, you can delete this object. You can also use the Radia System Explorer to delete the PACKAGE instance from the Radia Database.
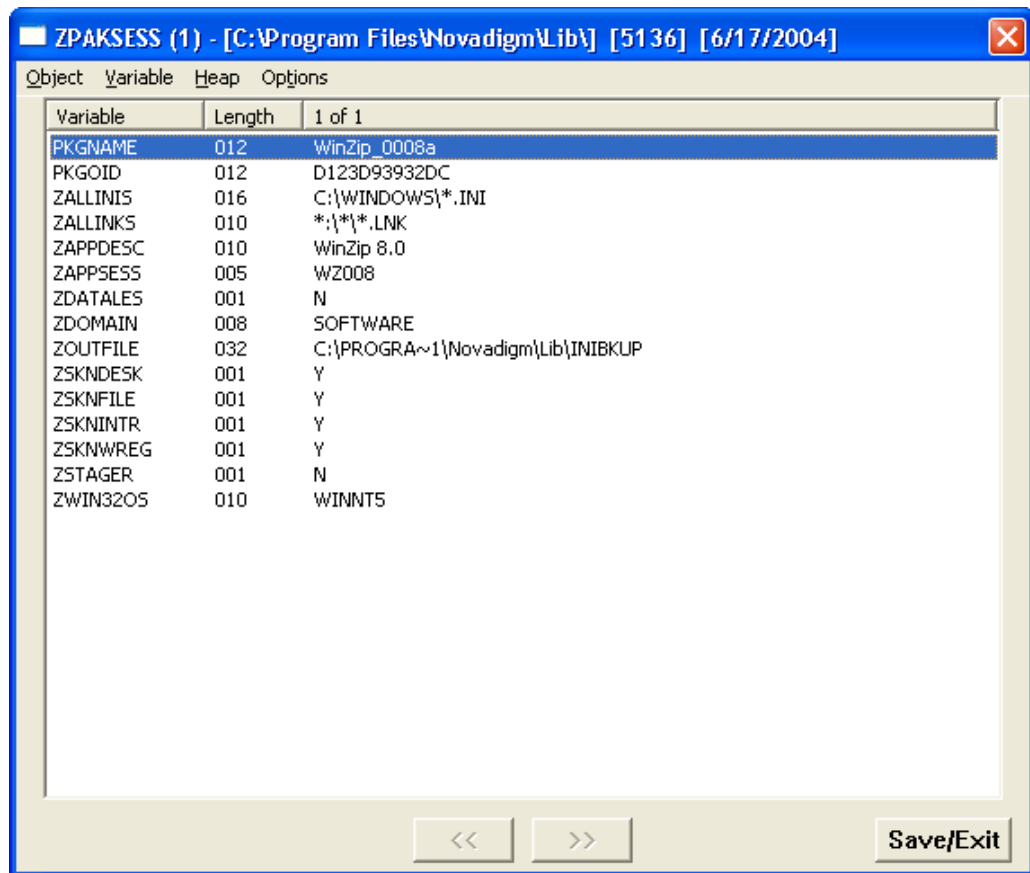
Figure 4.6 ~ ZPAKSESS.EDM, current session information (Installation Monitor Mode).

Click **Next** to go to the **System Configuration** window.

Step 4: Setting the Required System Configuration

Use the **System Configuration** window to limit the distribution of the package to computers that meet specific requirements. Distribution is based on the computer's operating system, installed RAM, and processor.
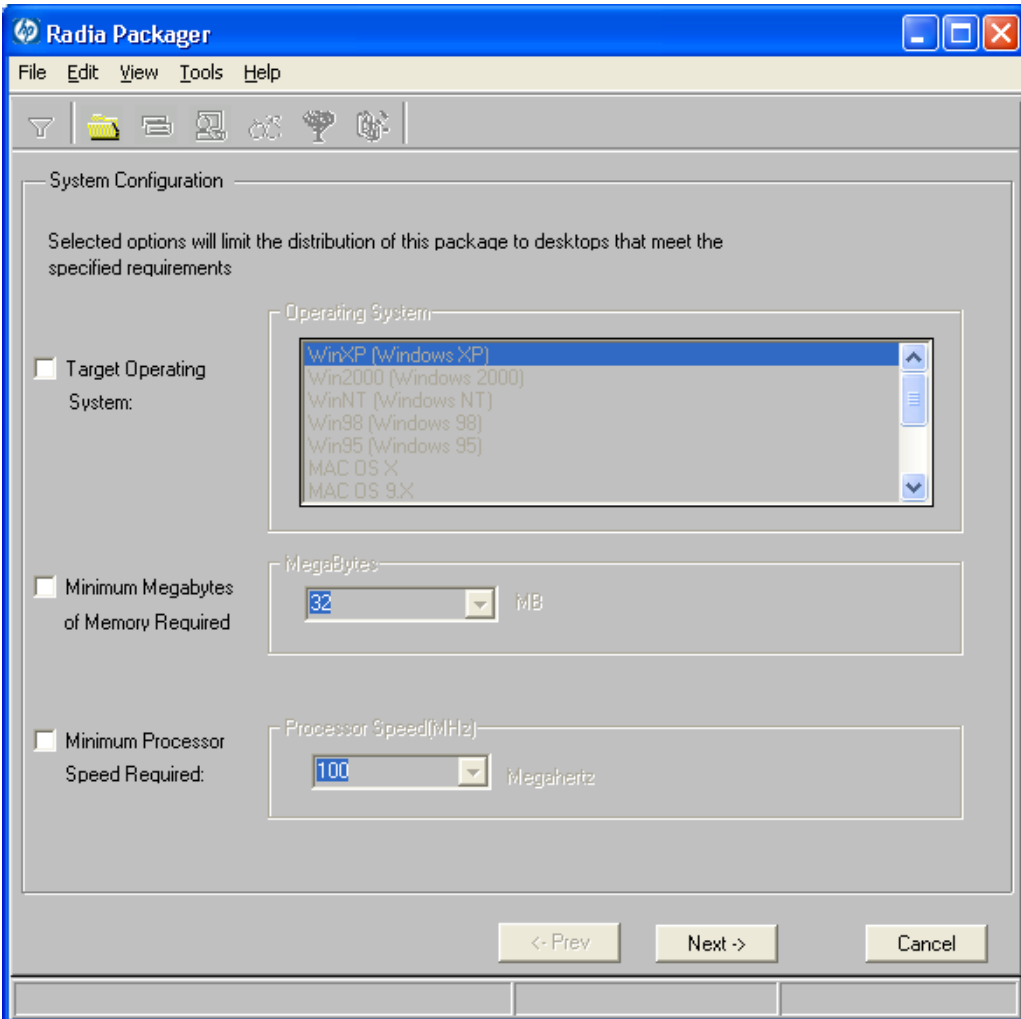
Figure 4.7 ~ System Configuration window (Installation Monitor Mode).

■ **Target Operating System**
Select this check box to make the **Operating System** list box available. Then, select one or more operating systems required for this package.

- To select multiple, consecutive operating systems, hold down the SHIFT key on your keyboard and click the appropriate items.

- To select multiple, non-consecutive operating systems, hold down the CTRL key on your keyboard and click the appropriate items. If no option is selected, the package will be available to subscribers running all operating systems.

■ **Minimum Megabytes of Memory Required**
Select this check box to make the **Megabytes** options available. Then, select the minimum number of megabytes of RAM required for this package. If no option is selected, the package will be available to subscribers with any amount of available memory.

■ **Minimum Processor Required**
Select this check box to make the **Processor** options available. Then, select the minimum processor that is required for this package. If no option is selected, the package will be available to subscribers with any processor. If none of the options is selected, the package will be available to all eligible subscribers.

In the WinZip example, as shown in Figure 4.7 on page 131, we accepted the defaults so as not to set any restrictions. The package can be delivered on computers running Windows 95, 98, NT, 2000, or Windows Server 2003.

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

Step 5: Setting Date and Time Constraints

Use the **Availability** window to specify the date and/or time when the package will be available for deployment. The date and time is based on the system clock, on the computer running the Radia Configuration Server. If no date and time constraints are specified, the package is available as soon as it is promoted to the Radia Database and configured for distribution.
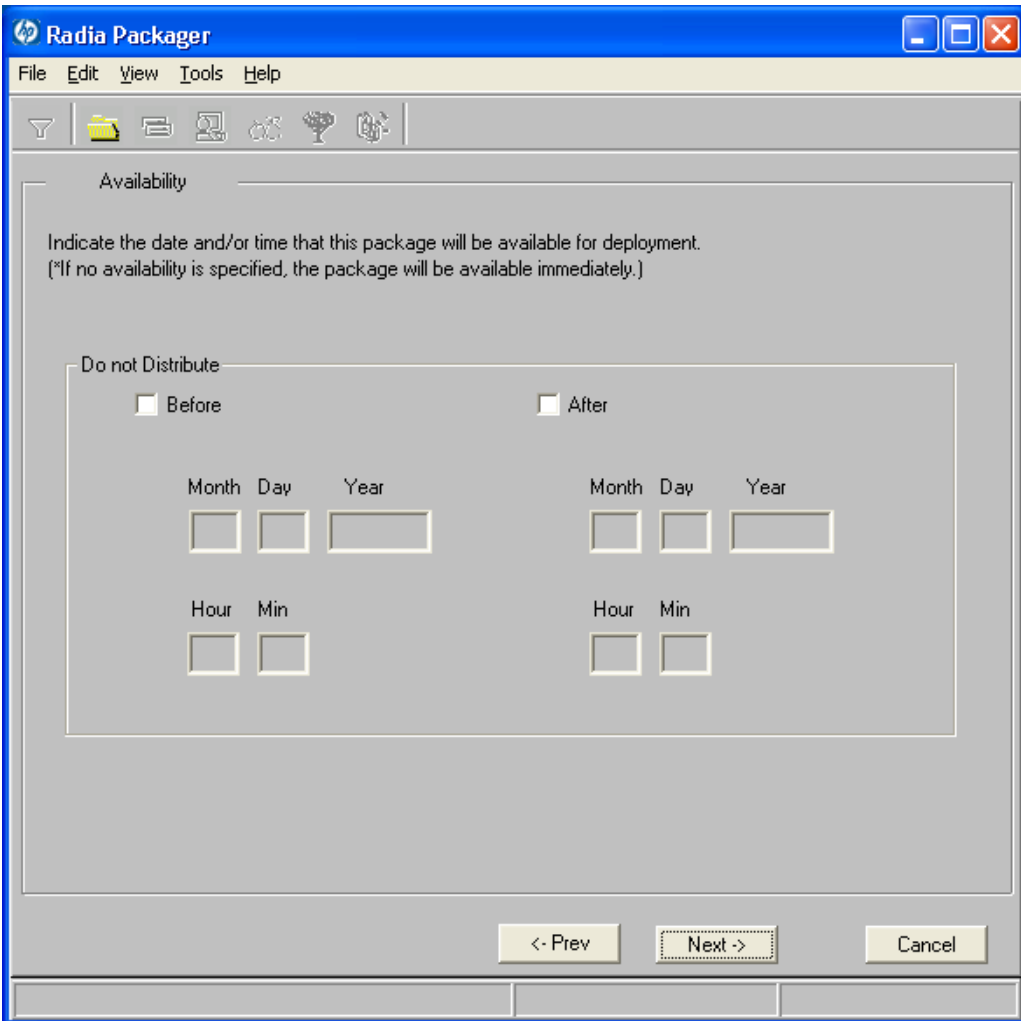
Figure 4.8 ~ Availability window (Installation Monitor Mode).

■ **Before**
Select this check box to prevent distribution of the package *before* the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.

■ **After**
Select this check box to prevent distribution of the package *after* the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.

In the WinZip example, as shown in Figure 4.8 on page 133, we left the default settings so that the package will be available for distribution as soon as we promote it to the Radia Database.

Click **Next** to go to the **What to Scan: Set Packaging Options** window.

Step 6: Identifying What to Scan

Use the **What to Scan: Set Packaging Options** window to identify areas where the software may make changes to the target computer. You can choose to scan the desktop, registry, file system, or to compare file content.
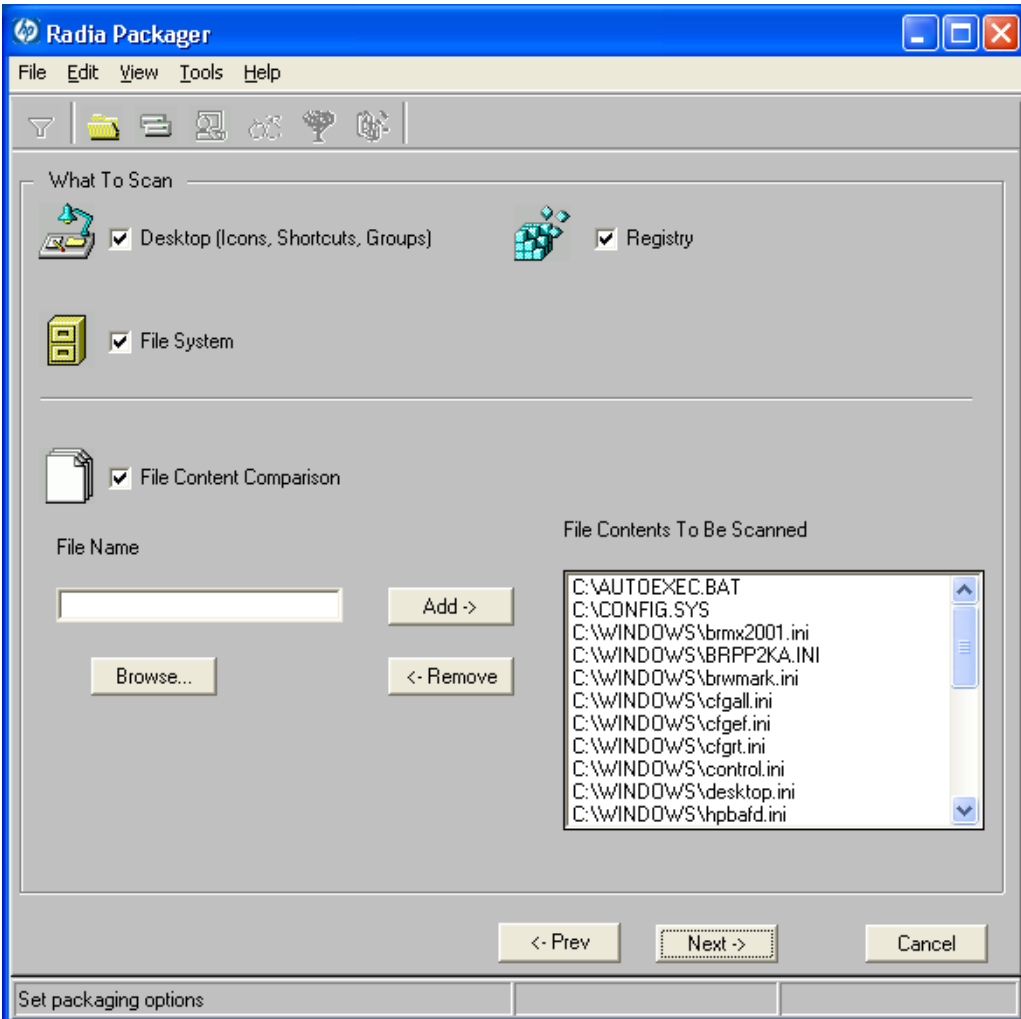
Figure 4.9 ~ Set Packaging Options: What to Scan window (Installation Monitor Mode).

Before you install the application, the Radia Packager scans the selected areas and takes a *snapshot* that records their current state. After you install the application, the Radia Packager

takes another *snapshot*. Radia compares these before and after snapshots, and the differences make up the package.

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

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

■ If you know that the application does not make any registry changes, you can clear the **Registry** check box so that Radia Packager does not scan the registry.

■ If you are packaging some favorites to Internet Explorer and you know that Internet Explorer stores favorites in individual .URL files, you can scan only the **file system** to pick up the changes.

■ If you are packaging several bookmarks to Netscape Navigator and you know that Netscape Navigator stores bookmarks in a single .HTM file, you can include this file in the **File Contents to be Scanned** list, select the **File Content Comparison** check box, and Radia Packager picks up the changes to the bookmark file.

You can see how the more information you have, the less time you will spend on scanning.

Note

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

To add files to be scanned

■ Type the full path and filename into the **File Name** text box, and then click **Add**, or click **Browse** to navigate to the appropriate file.

To remove files to be scanned

1. Click the file that you want to remove from the **File Contents to Be Scanned** list.

   OR

   If you want to remove multiple files from the list, press the CTRL key on the keyboard and click the files to remove.

2. Click **Remove**.

   In the WinZip example, as shown in Figure 4.9 on page 135, we accepted the default settings so that the entire desktop, registry, and file system will be scanned.

3. Click **Next** to go to the **Set Drives to Scan** window.

Step 7: Identifying What to Scan in the File System

Use the **Set Drives to Scan** window to select the areas of the file system to scan. All of the drives (local and network) connected to your computer are considered the *file system*. Many factors, including the computer's hardware and software, affect the amount of time it takes to scan the system.

Note
This window opens *only* if you selected the **File System** check box in the **Set Packaging Options: What to Scan** window.
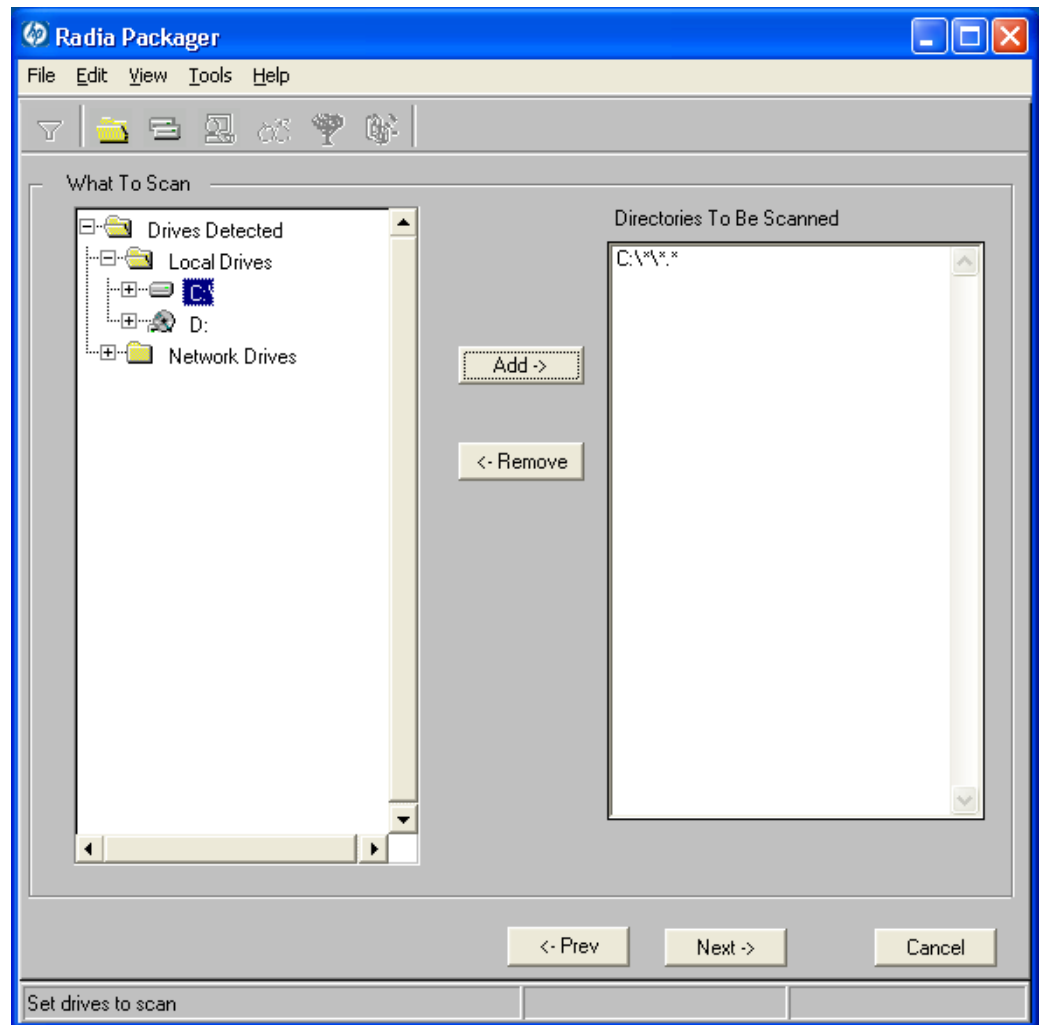
Figure 4.10 ~ What to Scan: Set drives to scan window (Installation Monitor Mode).

To set the drives to scan

1. Select *all* of the folders or drives that will be affected by the installation.

> **Caution**
>
> We recommend that you do *not* scan network drives. These can take longer due to the overhead transmission across the network. Also, the mapping of network drives may vary from one user to another.

2. Click **Add** to move your selections into the **Directories to be Scanned** list.

In the WinZip example, as shown in Figure 4.10 on page 138, we decided to scan the entire C drive for changes made when the application is installed.

Click **Next** to go to the **Pre-Installation Scan** window.

Step 8: Beginning the Pre-Installation Scan

Use the **Pre-Installation Scan** window to begin the scan. A check mark appears next to each area as the scan is finished. The Radia Packager alerts you when the scan is done.
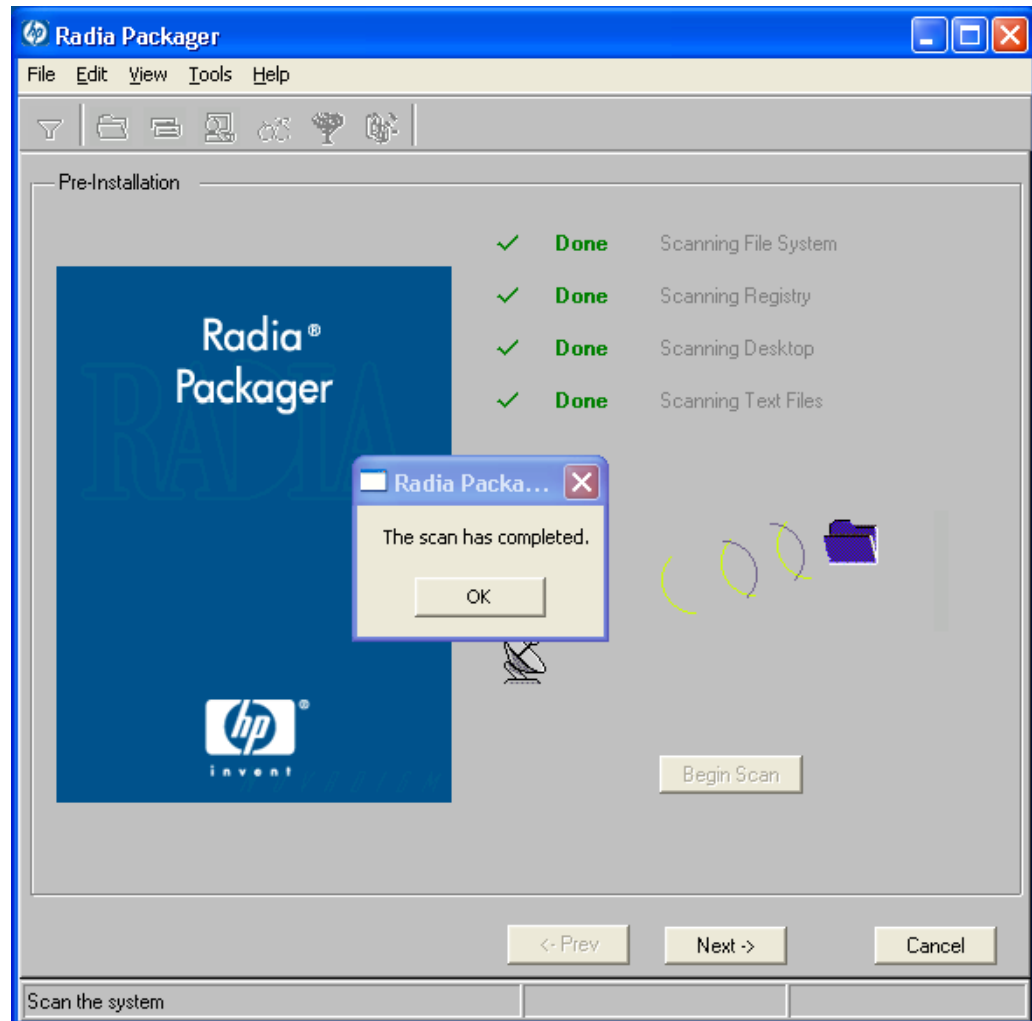
Figure 4.11 ~ Pre-Installation Scan window (Installation Monitor Mode).

To begin the pre-installation scan

1. Click **Begin Scan**. When the scan is done, a message box opens.
2. Click **OK**.

Step 9: Installing the Software

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

To install the software

1. In the **Pre-Installation Scan** window, click **Next**. The following message opens.
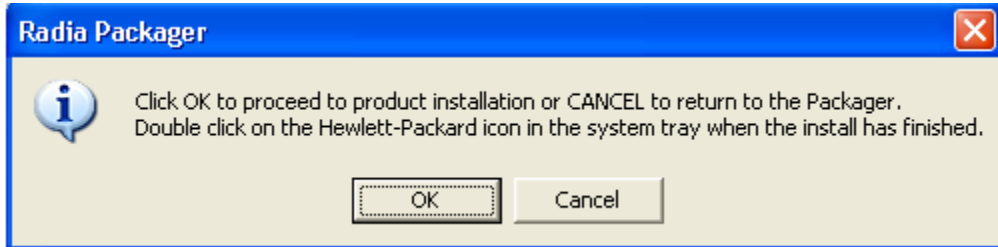
Figure 4.12 ~ Message to install the software (Installation Monitor Mode).

2. Click **OK**. The Radia Packager minimizes and the HP logo appears in the system tray.
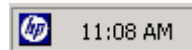
Figure 4.13 ~ HP logo in the system tray (Installation Monitor Mode).

3. Install the application using its native installation.

Caution
Do *not* do anything other than install the application. If you do, you risk making changes to the file system, registry, desktop, and so on. Radia will detect these (unrelated) changes during the post-installation scan and include them in the package.

Note
If the application that you are installing requires that you restart your computer, the HP logo in the system tray will be available when you log on again.

4. When the application is installed, double-click the HP logo in the system tray. A message asks if the installation was successful.

5. Click **Yes**. You may be prompted to type your **User ID** and **Password**.

In this example, WinZip 8.0 was installed in **C:\Program Files\WinZip**.

Step 10: Beginning the Post-Installation Scan

Use the **Post-Installation Scan** window to begin the final scan. Then, Radia Packager compares the results of the pre- and post-installation scans and alerts you when it is done. After closing the alert, you will see information about the number of changes made to files, registry keys, icons, and links.
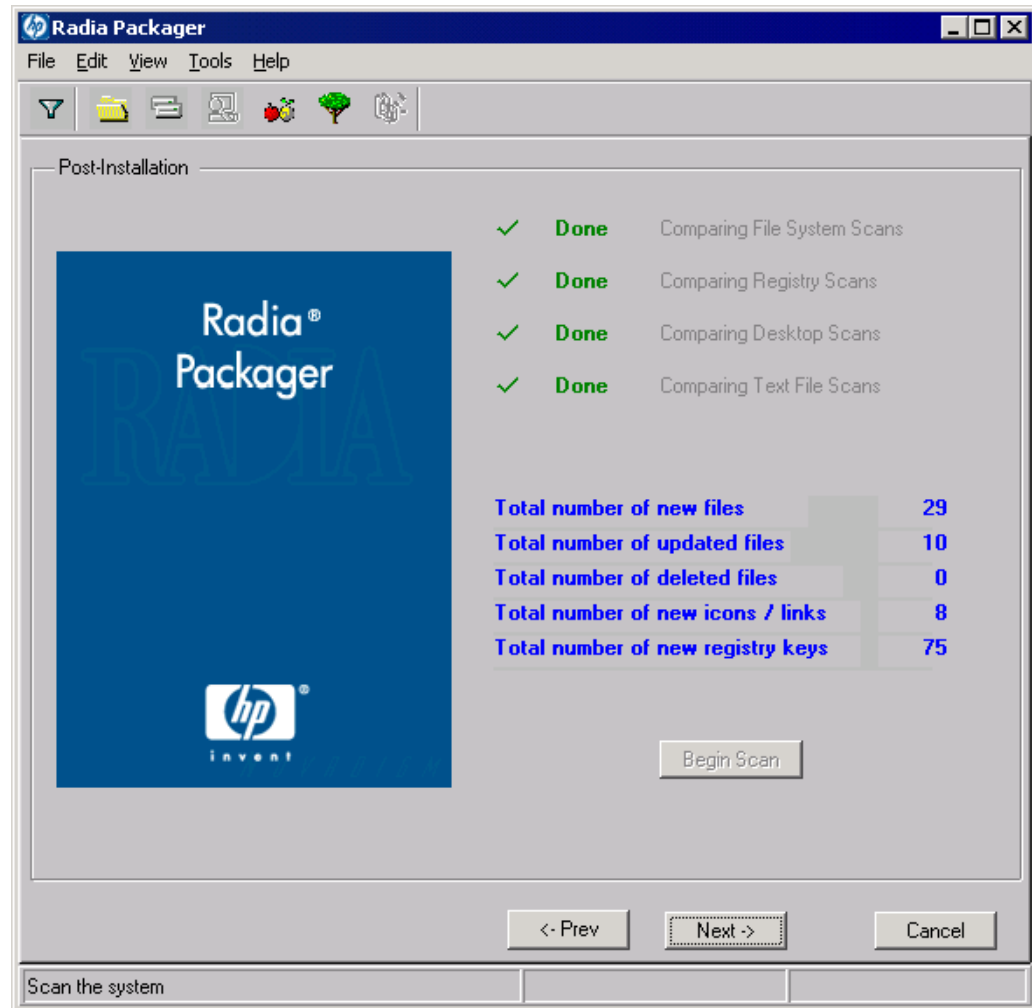
Figure 4.14 ~ Post-Installation scan window (Installation Monitor Mode).

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

To begin the post-installation scan

1. Click **Begin Scan**. When the scan is done, a message box opens.

2. Click **OK**.

> **Note**
>
> After differencing the before and after scans, the Radia Packager creates three registry (.EDR) files in the REGISTRY class. These files are promoted to the Radia Configuration Server.

Click **Next** to go to the **Summary** window.

Step 11: Reviewing the Modifications to the File System

Use the **Summary** window, shown in Figure 4.15 on page 147, to review the changes made when you installed the software to your computer. Three tabs are available: Files, Desktop, and Registry.

■   Click the **Files** tab to review the files that were added, deleted, or updated.

■   Click the **Desktop** tab to review the program groups, links, or shortcuts that were added, deleted, or updated.

■   Click the **Registry** tab to review changes detected in the registry.

Detailed information about the Files, Desktop, and Registry tabs follows, beginning with *Step 12: Viewing File Properties and Locations* on page 148.
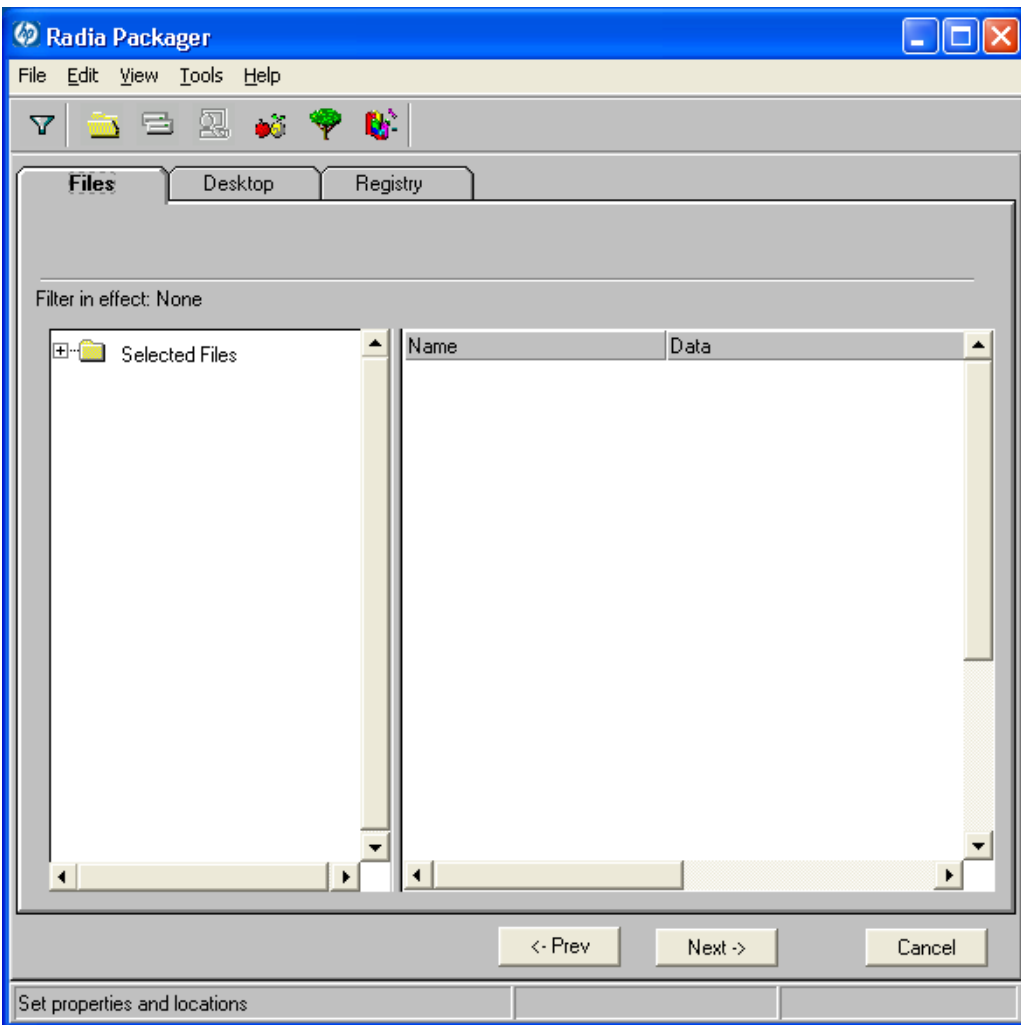
Figure 4.15 ~ Summary window (Installation Monitor Mode).

Step 12: Viewing File Properties and Locations

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

To view the selected files and folders

1. Right-click **Selected Files** and then select **Expand All**. Check marks indicate that the properties for the file have been specified and the Radia Packager is ready to promote them.
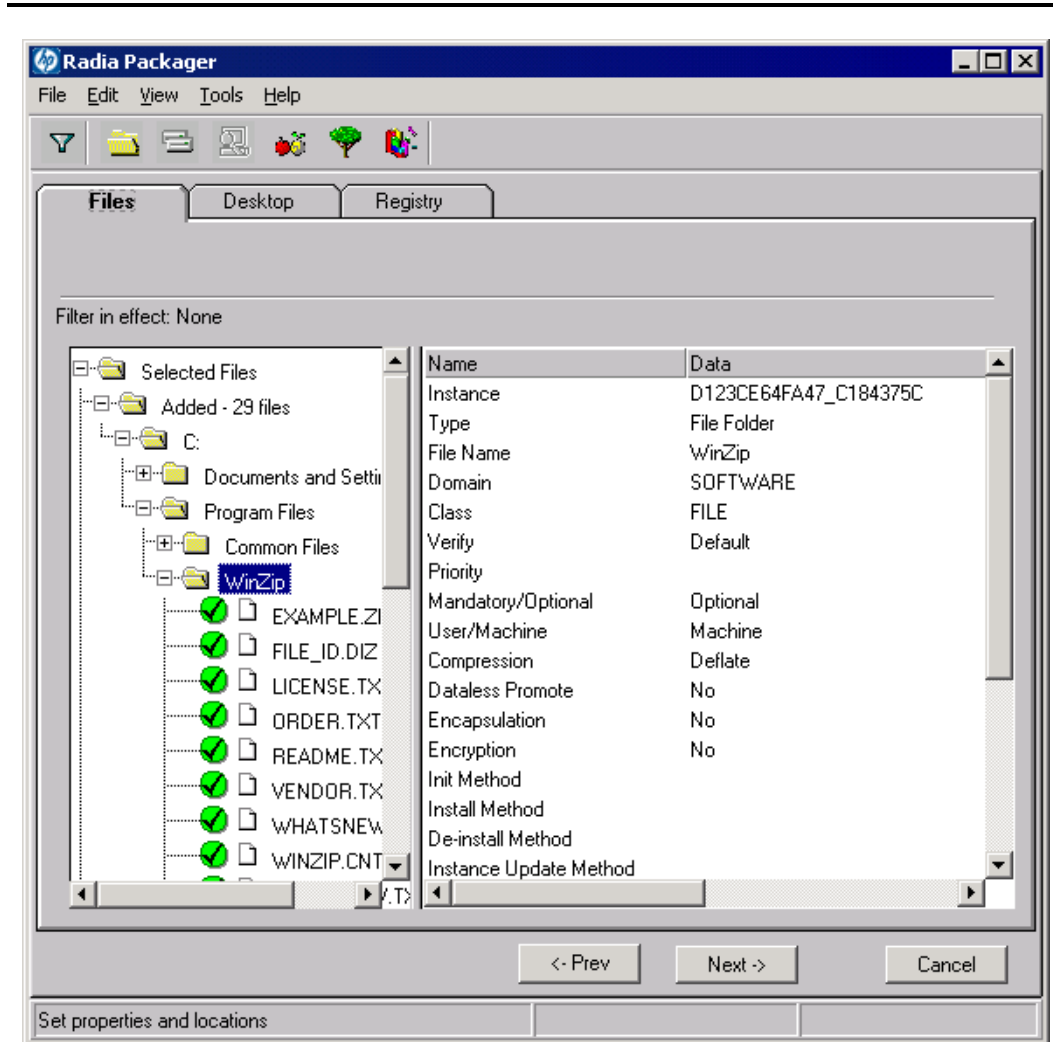
Figure 4.16 ~ View selected files (Installation Monitor Mode).

2. Select a file or folder in the tree view to see its properties in the list view.

In the WinZip example, we expanded the tree view to see all of the files and folders that are currently included in the package.

Step 13: Setting File Properties and Locations

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

To edit a file's properties

■ Right-click a file or folder. If you right-click a file, the following shortcut menu opens.
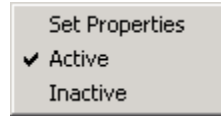
Figure 4.17 ~ Setting file properties (Installation Monitor Mode).

■ To exclude the file from the package, click **Inactive**. If you select a folder, all of the files in that folder will also be inactive.

 OR

■ To include the file in the package, click **Active**. This is the default setting. If you select a folder, all of the files in that folder will also be active.

 OR

■ To modify the properties of the file, select **Set Properties**. If you select a folder, you must also indicate whether you want the changes to affect the **Directory Only** or **Directory and Files**.
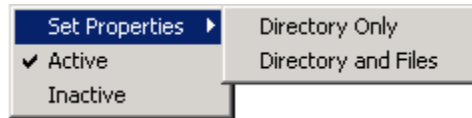
Figure 4.18 ~ Setting folder properties (Installation Monitor Mode).

The **Instance Properties** dialog box opens. Use this dialog box to modify the attributes of the files in the package. The properties of a file are stored in its instance in the Radia Database. The default values of these properties are set in the Global Default Properties dialog boxes. For more information see *Setting Default Packaging Properties* on page 221.

Client Management Tab

Use the Client Management tab to set verification and delivery options for the selected file or folder in the package. After promoting this package to the Radia Database, you can still modify these settings using the Radia System Explorer.

■ Use **Verification Options** to specify the verification actions that the Radia Software Manager will take for this file or folder.

■ Use **Delivery Options** to specify delivery options, such as the order in which files are deployed.
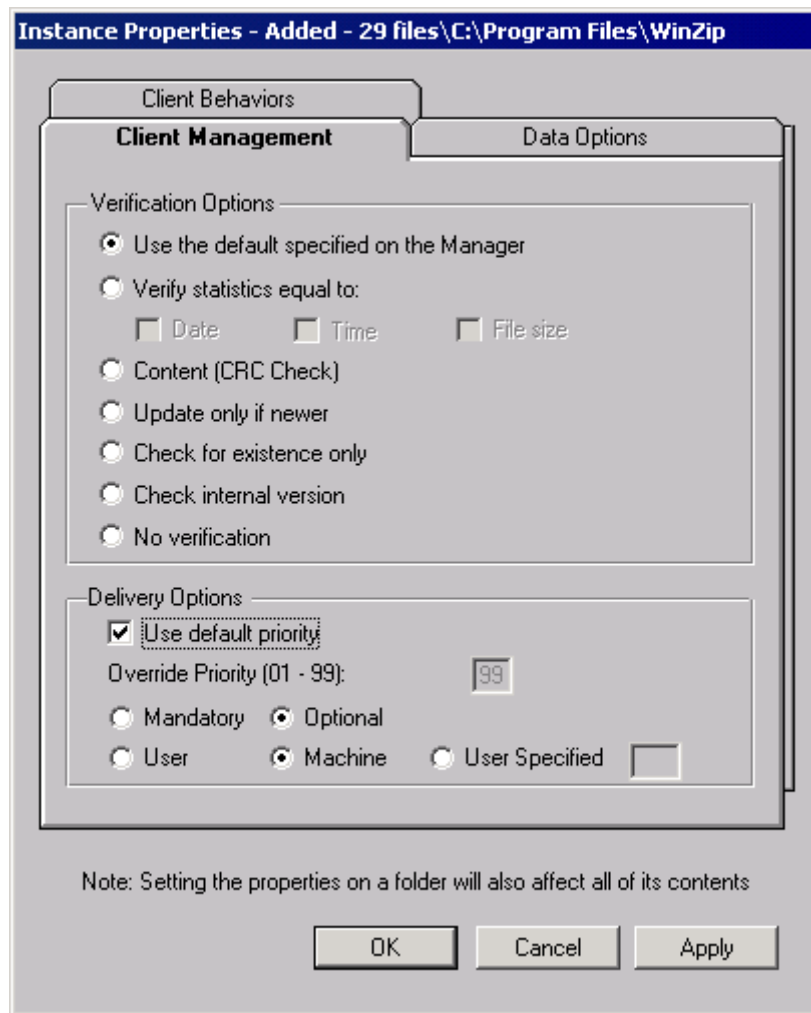
Figure 4.19 ~ Instance Properties dialog box, Client Management tab (Installation Monitor Mode).

VERIFICATION OPTIONS

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

Use the Radia System Explorer to look at the ZRSCVRFY attribute of the base instance of the FILE class to determine what the default verification options are.

■ **Verify statistics equal to**
Select this option so that the client checks the selected statistics (Date, Time, File Size) for the file on the client computer. The file is deployed from the Radia Database (or Radia Proxy Server) if its statistics are different from the file on the client computer. You can also use the Radia System Explorer to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.

■ **Content (CRC Check)**
Select the CRC option so that the Radia Software Manager checks the statistics Content (CRC) of the file on the client computer. When the file is verified, the file will *first* be checked based on the Date, Time, and File Size. To save time during the verify, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Radia Configuration Server (or Radia Proxy Server) will deploy a file if its statistics are different from those on the client computer. By selecting this option, the Radia Packager will calculate the CRC and put the value in the ZRSCCRC attribute.

■ **Update only if newer**
Select this option so that this file is deployed from the Radia Database (or Radia Proxy Server) if it has a later date/time stamp than the one on the client computer. You can also use the Radia System Explorer to set this option: ZRSCVRFY=U.

■ **Check for existence only**
Select this option so that this file is deployed only if it is not on the client computer. No action is taken if it already exists on the client computer, even if its statistics differ from those in the Radia Database. You can also use the Radia System Explorer to set this option: ZRSCVRFY=E.

■ **Check internal version**
Select this option to compare the version of the file being deployed against the version that may already exist on the client computer. This ensures that the correct file is being updated if more than one version exists. You can also use the Radia System Explorer to set this option: ZRSCVRFY=V. This feature applies to the following file types: DLL, .EXE, .OCX, .SYS, .VBX.

■ **No verification**
Select this option so that the file is deployed the first time the application is deployed. No subsequent action is taken. You can also use the Radia System Explorer to set this option: ZRSCVRFY=N.

DELIVERY OPTIONS

■ **Use default priority**
Select this check box to use the default priority of **50**. Priority determines the order of deployment, from highest priority to lowest priority. You can also use Radia System Explorer to set this option: ZRSCPRI=50.

■ **Override Priority (01-99)**
Type a number from **1** to **99** to override the default priority of **50**. **1** is the highest priority and **99** is the lowest. You can also use Radia System Explorer to set this option: ZRSCPRI=1.

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

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

● If there is not enough space on the client computer for the entire application, Radia will *only* deploy mandatory files.

● If there is not enough space for the mandatory files, then the application is not deployed at all.

You can also use the Radia System Explorer to set this option: ZRSCMO=M.

■ **Optional** (default)
Select this option to indicate that a file is not critical to the application.

● If there is not enough space on the client computer for the entire application, Radia will *not* deploy optional files.

You can also use the Radia System Explorer to set this option: ZRSCMO=O.

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

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

■ **Machine**
Select **Machine** to indicate that the file will be deployed to all users of the computer. You can also use the Radia System Explorer to set this option: ZCONTEXT=M.

■ **User Specified**
This option is for future use.

In the WinZip example, we accepted the default verification and delivery settings.

Data Options Tab

Use the Data Options tab to specify data compression and other details about the files or folders that you will be distributing.
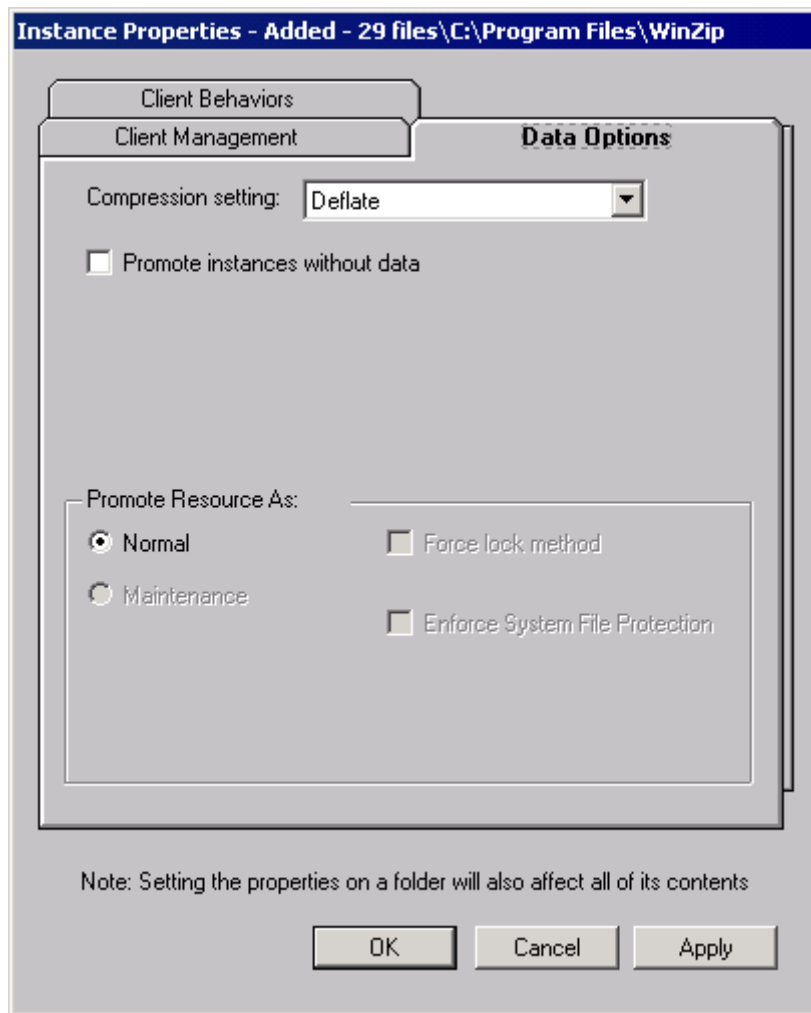
Figure 4.20 ~ Instance Properties dialog box, Data Options tab (Installation Monitor Mode).

■ **Compression setting**
If necessary, select the compression setting for storing this file in the Radia Database.

Compression minimizes the time required to transmit data and the amount of disk space required to store it.

- Select **Deflate** for the most efficient compression, which produces smaller compressed images.

- Select **None** if the file is already compressed. If you are packaging an application that contains one or more compressed files, do *not* have Radia Packager compress the file as well. The file may actually grow in size if it is compressed again.

■ **Promote instances without data**
Select this check box to indicate that this file should not be transferred to the Radia Database as part of the package. Only the instance representing it is included in the package. The data remains in compressed format in the IDMDATA location on your computer. The default IDMDATA location is *SystemDrive***:\Program Files\Novadigm\Lib\Data**. You can manually place a file on a Radia Staging Server if you have connectivity to it and do not want a copy of it in the Radia Database.

Caution
We recommend that you do *not* use this property without a thorough understanding of the extended security in the NT operating system.

PROMOTE RESOURCE AS

■ **Normal**
Select this option to indicate that this file is to be deployed as part of an application.

■ **Maintenance**
This is a feature from previous versions and, as of version 4.0, is no longer used.

■ **Force lock method**
Select this check box to force the use of the *locked file method* for deploying this file. If the file is in use on the client computer when Radia attempts to deploy a new copy of it, the locked file method is normally used to deploy it. If necessary, it is decompressed and stored locally in a folder. The Client Connect process forces a restart when it ends and the file is deployed to its correct location during startup.

■ **Enforce System File Protection (Windows 2000 and Windows Server 2003 only)**
Select this check box to specify that you want to enforce Windows System File Protection (SFP). Radia first attempts to verify the resource using the current verify options, as specified in the ZRSCVRFY attribute located in the FILE class.

● If the verification is successful, the resource is in the desired state and no further verification is necessary.

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

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

● If you select this check box, the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing.

● If you do not enforce SFP, the file is marked as having passed verification.

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

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

Client Behaviors Tab

Use the Client Behaviors tab to specify methods (programs) that Radia executes on the client computer.
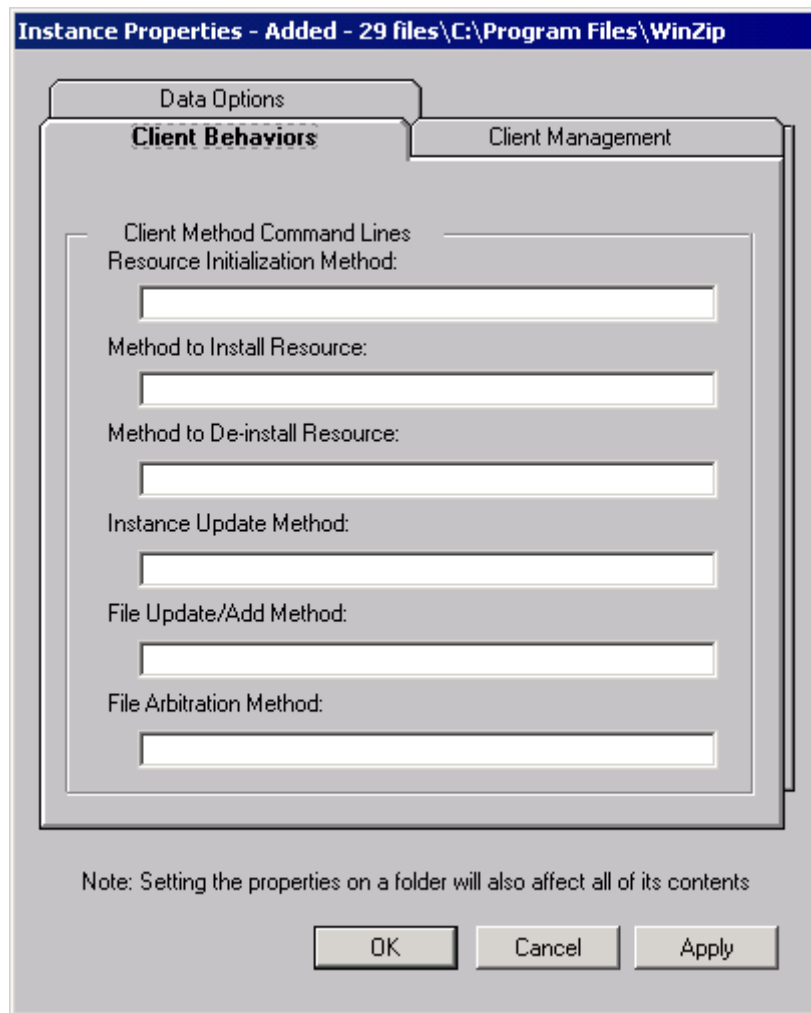
Figure 4.21 ~ Instance Properties dialog box, Client Behaviors tab (Installation Monitor Mode).

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

CLIENT METHOD COMMAND LINES

■ **Resource Initialization Method** (Attribute in Database: ZINIT)
Type the method to run when the file is stored on the client computer.

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

■ **Method to De-install Resource** (Attribute in database: ZDELETE)
Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared DLL file, is not supposed to be deleted from the client computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for the **Method to De-install Resource**.

■ **Instance Update Method** (Attribute in database: ZUPDATE)
Type the method to run when the instance is modified on the computer, after the file has been deployed.

■ **File Update/Add Method** (Attribute in database: ZFILEUPD)
Type the method to run when the file is new or has been updated. The method executes just before it is deployed to the client computer.

■ **File Arbitration Method** (Attribute in database: ARBITRAT)
Type the method to run if a file is about to be replaced. This method compares the version information of the file or folder on the client computer and the one that is going to replace it, and then determines which to keep.

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

In the WinZip example, we have not specified any methods.

Step 14: Working with Updated Files

In addition to the files that were added, you might notice *Updated files*, as shown below.
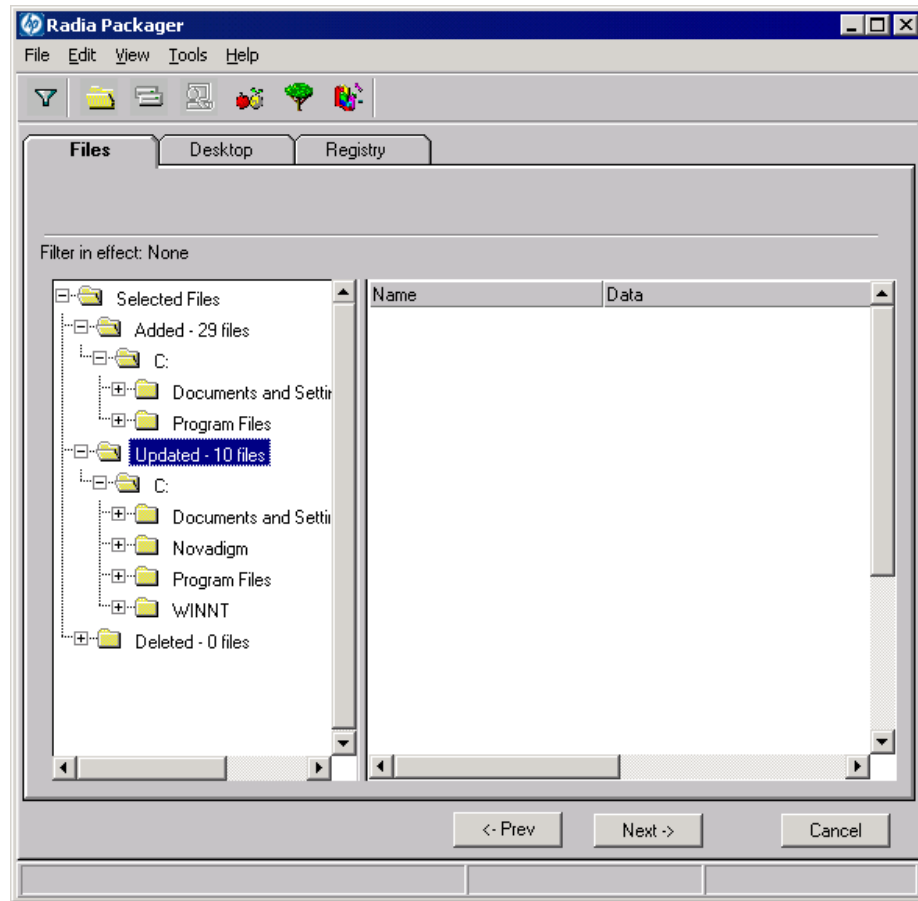
Figure 4.22 ~ View updated files (Installation Monitor Mode).

Updated files are control files whose ASCII text has been modified, such as Win.ini in the image above. You can set the properties for a standard ASCII text file as described in *Step 13: Setting File Properties and Locations* on page 150.

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

To edit a line in a control file

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

■   Click **Active** to include the change to the text file in the package.

    OR

■   Click **Inactive** if you do *not* want to include the change to the text file in the package.

    OR

■   Click **Override Value** if want to modify the change to the text file.

    If you selected **Override Value**, the **Override Text File Data** dialog box opens.
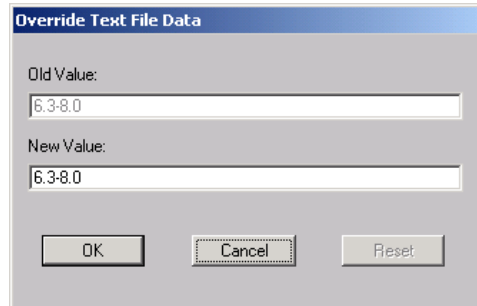
Figure 4.23 ~ Override Text File Data dialog box (Installation Monitor Mode).

To use the Override Value feature

1.   Click **Override Value**.

2.   Type the **New Value** for the line in the text box.

3.   Click **OK**.

To restore the text line

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

2.   Click **Override Value**.

3.   Click **Reset**.

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

In the WinZip example, we set the changes listed to the Modem log to inactive (see Figure 4.22 on page 160). However, we did not make any modifications to the lines in the control file.

Step 15: Setting Desktop Properties and Locations

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

To view the selected program groups, links, and shortcuts

■   Right-click **Selected Links,** and select **Expand All**.
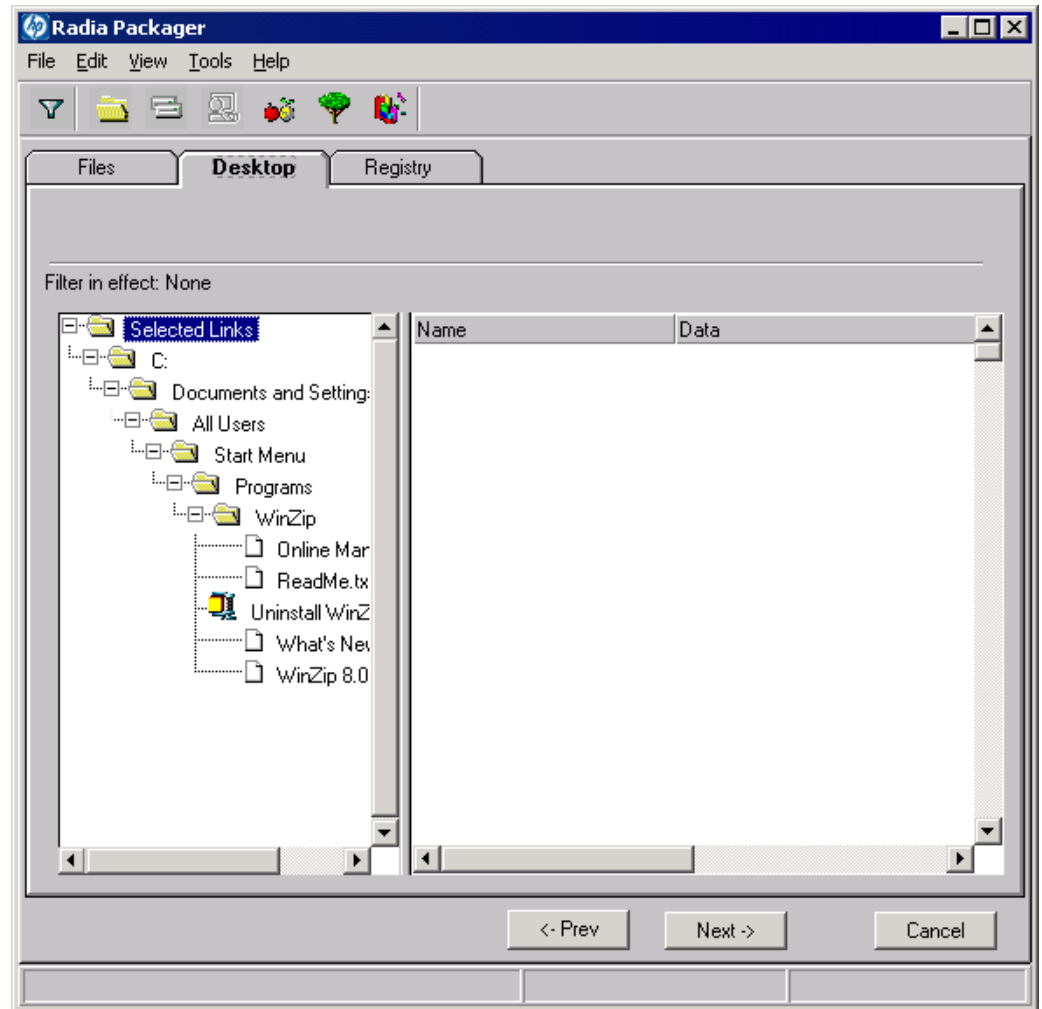
Figure 4.24 ~ View selected program groups, links, and shortcuts (Installation Monitor Mode).

To set properties for program groups, links, and shortcuts

■　Right-click the link in the tree view. A menu opens.
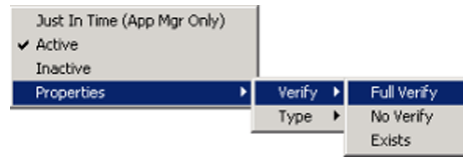
Figure 4.25 ~ Expanded shortcut menu (Installation Monitor Mode).

> **Note**
>
> When you right-click a folder or link, Radia applies the changes to all links within the folder or within the link's branch in the tree view.

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

■　**Properties**
Select **Properties** to open the **Verify/Type** submenu.

Verify Menu Commands

■　**Full Verify**
Select this command to deploy the link every time a subscriber connects to the Radia Configuration Server, even if it already exists on the client computer. This repairs broken or modified links.

■　**No Verify**
Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.

■　**Exists**
Select this command to deploy the link only if it does *not* exist on the client computer, during initial deployment and on each subsequent connection. The link is deployed again only if it was deleted. This allows users to modify the link.

Type Menu Commands

The following options apply only to operating systems supporting multiple users, such as Windows 95, 98, NT 4.0, 2000, or Windows Server 2003.

■ **User**
   Select this command to deploy this link only to the subscriber logged on when the application is initially deployed.

■ **Machine**
   Select this command to deploy this link to all users of the computer.

In the WinZip example, we left the default settings for program groups, links, and shortcuts.

Step 16: Setting Registry Properties and Values

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

To view the selected registry

■   Right-click **Selected Registry Keys,** and select **Expand All**.
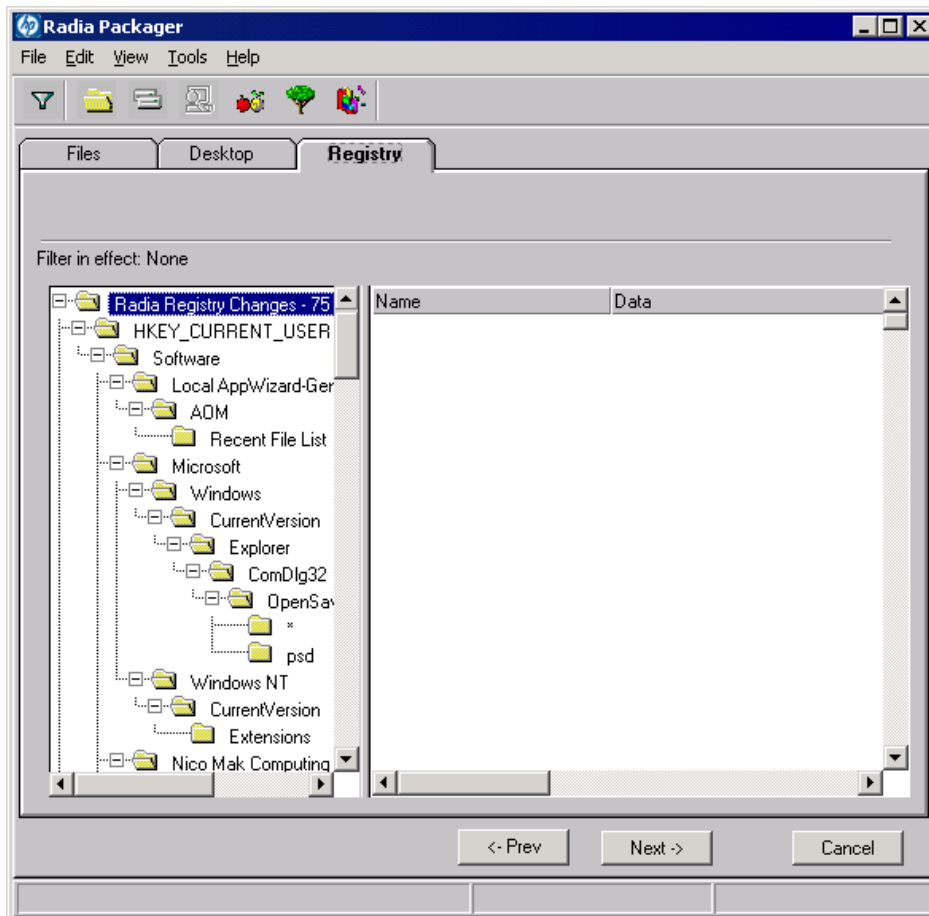
Figure 4.26 ~ View selected registry keys (Installation Monitor Mode).

To edit values in a registry key

1. Select the registry key in the tree view.
2. Double-click the registry key in the list view.
3. Edit the data in the **New Value** text box.
4. Click **OK** to save your changes and close the dialog box.
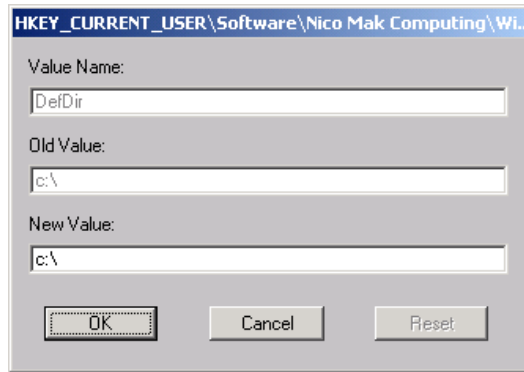
Figure 4.27 ~ New Value text box (Installation Monitor Mode).

To restore the registry key to its original value

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

To set properties for registry keys

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

Figure 4.28 ~ Set Properties for registry keys (Installation Monitor Mode).

> **Note**
>
> When you right-click a registry key, Radia applies the changes to all registry keys within the key's branch in the tree view.

- **Active**
  Select this command to include registry changes in the package.

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

- **Properties**

  ♦ **Unconditional/Enforced**
    Select this command to deploy the registry key, even if it already exists on the client computer. Every time the subscriber connects to the Radia Configuration Server, this registry key is reset to the packaged value, if it has been modified on the client computer. Use this setting for registry keys that must not change in order for the application to function properly.

  ♦ **Unconditional/Exist** (default)
    Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset.

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

In the WinZip example, we set the registry keys to **Unconditional/Exist**. Click **Next** to go to the **Promote Files** window.

Step 17: Promoting Packages

Use the **Promote Files** window to create instances for the package in the Radia Database.

Caution

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

Copy the LIB directory (and its subdirectories) from ***SystemDrive*:\Program Files\Novadigm\LIB** to a backup location. The LIB directory contains the objects that were built by the Radia Packager during your packaging session.

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

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

To view the files to be promoted

■   Right-click **Files to be Promoted**, and select **Expand All**.
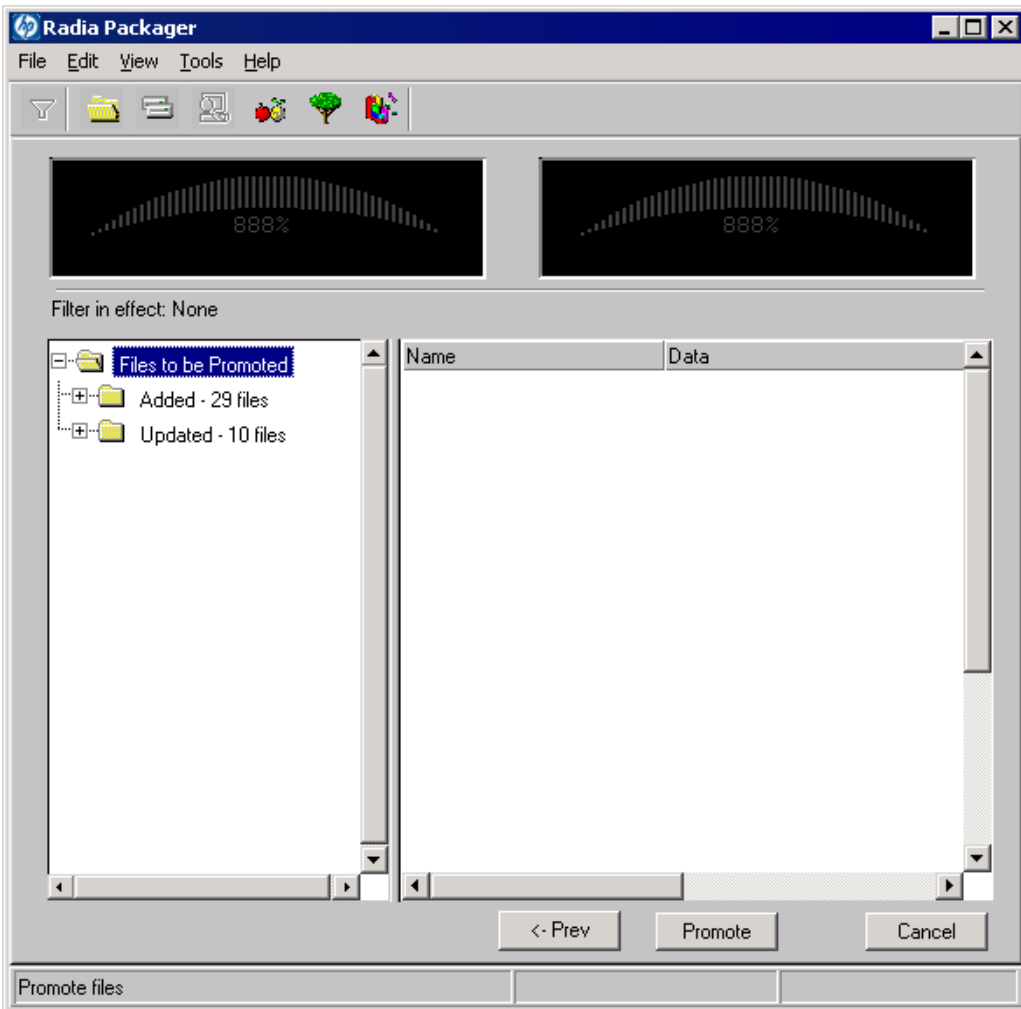
Figure 4.29 ~ Promote Files window (Installation Monitor Mode).

To promote the package

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

In this example, WinZip 8.0 was promoted to the Radia Database. Next, use the Radia System Explorer to create a service. For more information, see *Creating a Service* on page 212.

Using Component Selection Mode

In Component Selection Mode, you select the individual components that make up the application, such as files, directories, registry entries, and links to create a package. We recommend this mode for packaging simple applications.

> **Note**
>
> If the application makes changes to an existing text file, such as Win.ini, you *must* use Installation Monitor Mode.
>
> Also, use Installation Monitor Mode to package any application with which you are not completely familiar.

Packaging in Component Selection Mode involves three phases:

1. **Defining** the application's hardware, software, and operating system requirements.

2. **Selecting** the individual files to be packaged.

3. **Promoting** the files to the Radia Database.

Prerequisites

Before packaging your application in Component Selection Mode:

■ Install the target application on your (administrator) computer. This ensures that the files you need to select reside on the computer.

■ If the application makes changes to the registry, export the registry entries.

> **Caution**
>
> When you export the registry entry, be sure that you set the Export range to **Selected branch** so that you do not export all registry entries.
>
> Also, if you are running Windows 2000 or Windows Server 2003, select **Win9x/NT 4 Registration Files (REGEDIT4)** from the **Save as Type** menu.

Packaging with Component Selection Mode

This section guides you through packaging a sample application using the Component Selection Mode and provides detailed information about each screen that you encounter.

In the example, you will package GS-CALC, a simple shareware application available at http://www.jps-development.com/gs-calc.htm. The target platform is Windows 2000.

Use this example to familiarize yourself with Component Selection Mode. However, please remember that there are many variables when packaging applications.

Reminders

For the following example to work, be sure to:

- Install GS-CALC on your administrator computer.
- Export the registry entry in **HKEY_CURRENT_USER\SOFTWARE\JPS Development**.

Step 1: Logging On to Radia Packager

1. Go to **Start**, **Programs**, **Radia Administrator Workstation**, and select **Radia Packager**.

2. In the **Radia Packager Security Information** dialog box, type your **User ID** and **Password** in the appropriate text boxes.

Note

The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

---

3. Click **OK**.

   The **Open Packaging Session** window opens.

Step 2: Completing the Open Packaging Session Window

Use the **Open Packaging Session** window to select the packaging mode and enter information to identify the packaging session.
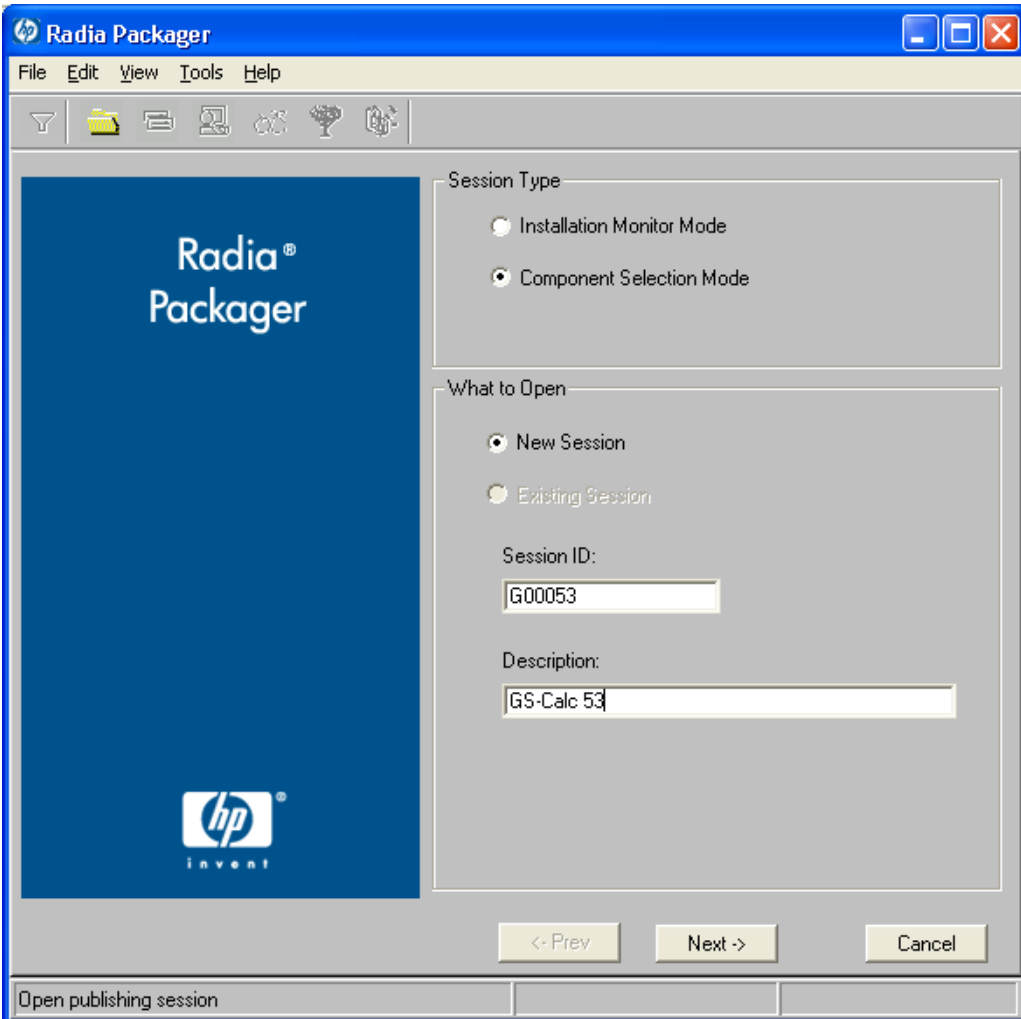
Figure 4.30 ~ Open Packaging Session window (Component Selection Mode).

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

The following sections describe the options on the **Open Packaging Session** window, as shown in Figure 4.30 on page 173.

> **Note**
>
> Review the Global Defaults prior to beginning any Packaging Session. See *Setting Default Properties* beginning on page 221.

Session Type Area

■ **Installation Monitor Mode**
Use this mode to create Radia packages when you do not know the structure of the application or data. Radia Packager performs a scan of your computer, and then prompts you to install the software using the native installation program. After the software is installed, Radia Packager performs another scan and then compares the scans. The differences are packaged and promoted to the Radia Database. For more information, see *Using Installation Monitor Mode* on page 123.

■ **Component Selection Mode**
Use this mode to create Radia packages when you know which files need to be distributed and what impact the installation will have on the client computer. In this mode, you identify the files that are packaged and then promoted to the Radia Database.

> **Note**
>
> In previous versions of Radia Packager, Import Mode was available for applications installed or maintained using Windows Installer. For information on packaging these applications, see the *Radia Extensions for Windows Installer Guide*. This is a separately licensed product.

What to Open Area

■ **New Session**
Select **New Session** to begin a new packaging session. You must complete the **Session ID** and **Description** text boxes.

■ **Existing Session**
Select **Existing Session** to resume the previous session.

■ **Session ID**
Type a *unique* identifier that is one to six characters long.

■ **Description**
Type a description of the session.

In the GS-CALC example, as shown in Figure 4.30 on page 173, we are using **Component Selection Mode** to begin a **New Session**. The **Session ID** is **G00053** and we've described the session as **GS-Calc 53**.

Click **Next** to go to the **Package Properties** window.

Step 3: Entering Package Properties

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

Figure 4.31 ~ Package Properties window (Component Selection Mode).

- ■ **Package Name**
  Type a name for the package. This is the name for the PACKAGE class instance in the Radia

Database and should conform to your naming conventions. Note that the name cannot contain any spaces.

■ **Domain**
Select the domain in which to store the instance. This is normally the SOFTWARE domain unless you have customized the Radia Database with proprietary domains..

■ **Description**
Type a description for the package.

■ **Release**
Type the release number of the software.

In the GS-Calc example, as shown in Figure 4.31 on page 176, we named the package **GSCalc_00053** and gave it a description and release number, **GS-Calc,** and **5.3** respectively.

Note

The current session information is stored in ***SystemDrive*:\Program Files\Novadigm \LIB\ZPAKSESS.EDM** on your computer. Use the Radia Client Explorer to view this object.

Figure 4.32 on page 178 contains an example of ZPAKSESS.EDM.

If you decide to close this packaging session and do not plan to complete it, you can delete this object. You can also use the Radia System Explorer to delete the PACKAGE instance from the Radia Database.
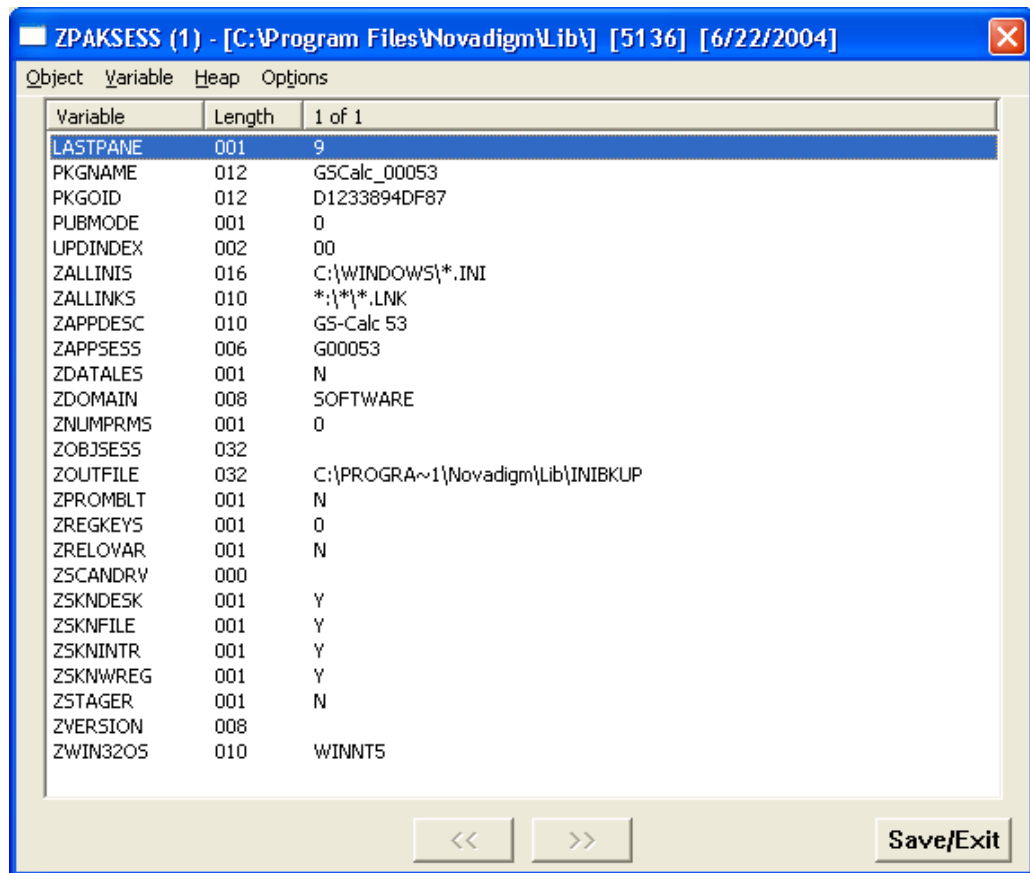
Figure 4.32 ~ ZPAKSESS.EDM, current session information (Component Selection Mode).

Click **Next** to go to the **System Configuration** window.

Step 4: Setting the Required System Configuration

Use the **System Configuration** window to limit the distribution of the package to computers that meet specific requirements. Distribution is based on the computer's operating system, installed RAM, and/or processor. If none of the options is selected, the package will be available to all eligible subscribers.
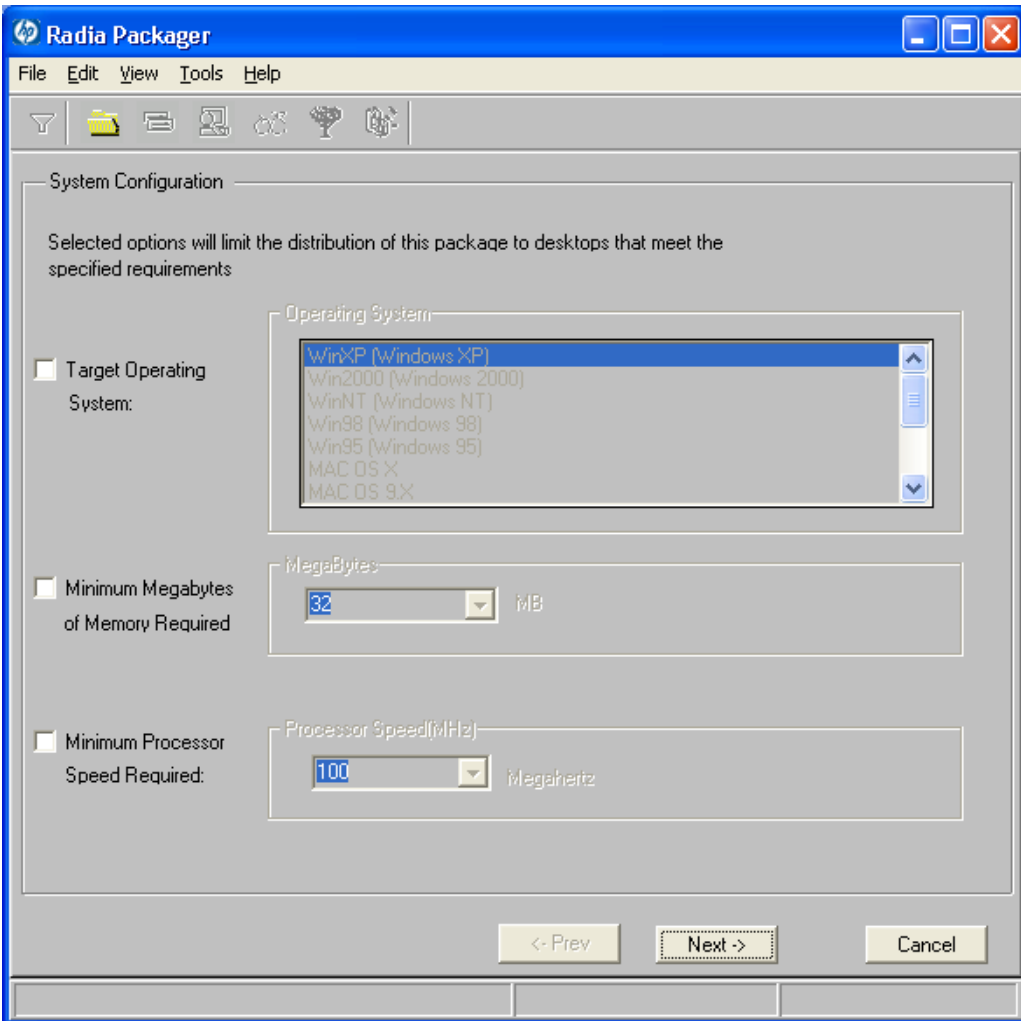
Figure 4.33 ~ System Configuration window (Component Selection Mode).

■ **Target Operating System**
Select this check box to make the **Operating System** list box available. Then, select one or more operating systems required for this package.

- To select multiple, consecutive operating systems, hold down the SHIFT key on your keyboard and click the appropriate items.

- To select multiple, non-consecutive operating systems, hold down the CTRL key on your keyboard and click the appropriate items. If no option is selected, the package will be available to subscribers running all operating systems.

■ **Minimum Megabytes of Memory Required**
Select this check box to make the **Megabytes** options available. Then, select the minimum number of megabytes of RAM required for this package.

■ **Minimum Processor Required**
Select this check box to make the **Processor** options available. Then, select the minimum processor that is required for this package.

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

Step 5: Setting Date and Time Constraints

Use the **Availability** window to specify the date and/or time when the package will be available for deployment. The date and time is based on the system clock, on the computer running the Radia Configuration Server. If no date and time constraints are specified, the package is available as soon as it is promoted to the Radia Database and configured for distribution. After promoting this package to the Radia Database, you can still modify these settings using the Radia System Explorer.
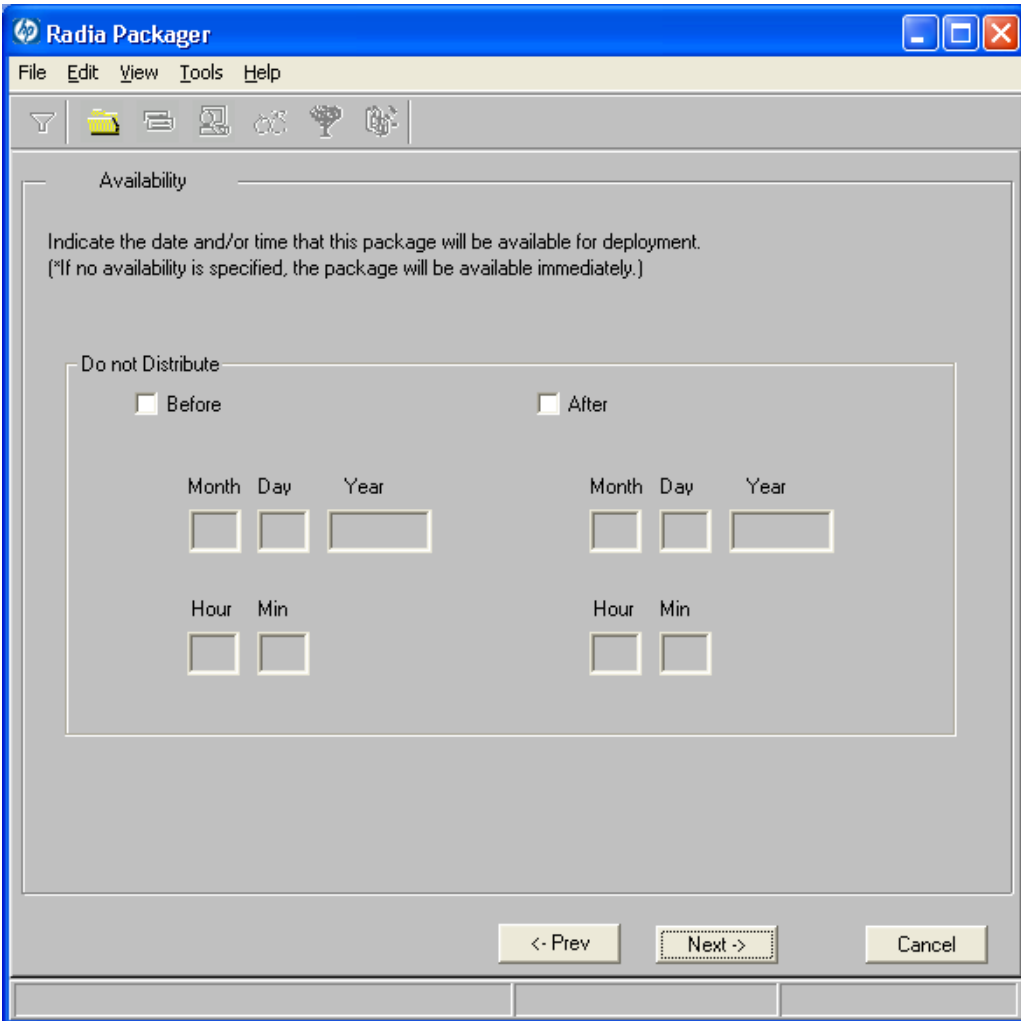
Figure 4.34 ~ Availability window (Component Selection Mode).

■ **Before**
   Select this check box to prevent distribution of the package *before* the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.

■ **After**
   Select this check box to prevent distribution of the package *after* the specified date and time. Use the **Month**, **Day**, **Year**, **Hour**, and **Min** text boxes to specify the date and time.

In the GS-Calc example, as shown in Figure 4.34 on page 181, we accepted the default settings so that the package will be available for distribution as soon as we promote it to the Radia Database.

Click **Next** to go to the **Select Files to Be Packaged** window.

Step 6: Selecting the Files to Package

Use the **Select Files to be Packaged** window to select all files, links, or shortcuts, and registry entries that need to be included in the package. If you are unsure about any of these components, consider using Installation Monitor Mode instead.

To select the files to package

■ Navigate through your file system (shown in the tree view in Figure 4.35 on page 184) and select the files or folders to be included in the package.

Click a check box again to clear a selection.
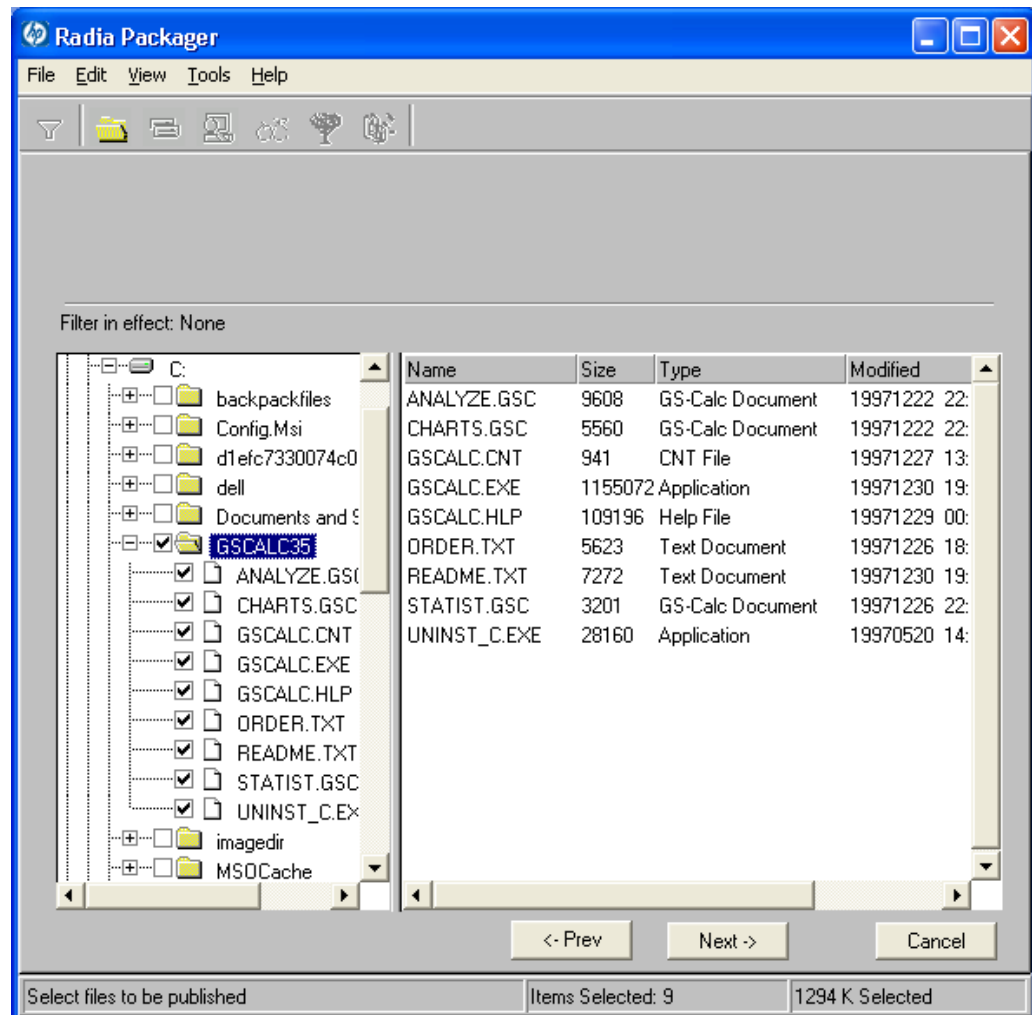
Figure 4.35 ~ Select Files to be Packaged window (Component Selection Mode).

In the GS-Calc example, we selected the folder **C:\GS-Calc** that contains the program files, as shown in Figure 4.35 above. Next, we selected the links located in the folder **C:\Documents and Settings\All Users\Start Menu\Programs\GS-Calc**. Finally, we selected the registry entry that we exported earlier.

Click **Next** to go to the **Set Properties and Locations** window.

Caution

When you add a new line to a batch or text file as part of a package, the packaging process uses the previous line as the search line when applying those changes during deployment. Be aware that if you have two identical lines, such as REM, the service will automatically put the new line after the first occurrence of REM.

Step 7: Viewing File Properties and Locations

Use the **Files** tab on the **Set Properties and Locations** window to see the selected files and folders in the package.

To view the selected files and folders

1. Right-click **Selected Files** and select **Expand All**. Check marks indicate that the properties for the file have been specified and the Radia Packager is ready to promote them.

2. Select a file in the tree view to see its properties in the list view.

Some of the properties in Figure 4.36 on page 187, such as Verify, Priority, and Mandatory/Optional, are initially set according to the selections in the **Radia Packager – Global Default Properties** dialog box, as described starting on page 221. In *Step 8: Setting File Properties and Locations* on page 189, you will learn how to modify these settings, if necessary.
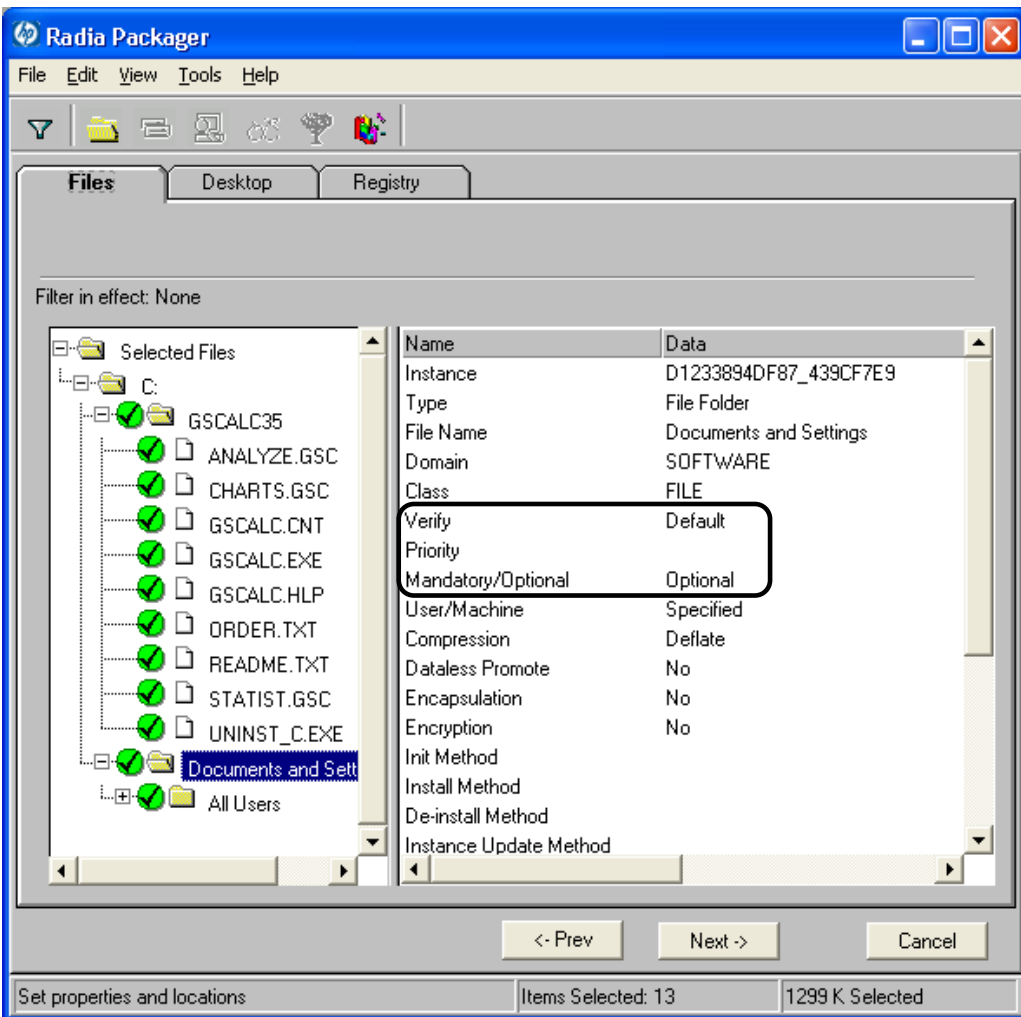
Figure 4.36 ~ View selected files window (Component Selection Mode).

To filter the displayed files and folders

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

1. Click the filter button ▽ to filter the files in the tree view.

2. In the **Set Filter** dialog box, type the file type as a string, such as **.DLL**. If the filter string is found anywhere in the file or folder name, it is considered a match.

3. Click **OK**.

To remove a filter

1. Click the filter button to open the **Set Filter** dialog box.

2. Click **Clear**.

3. Click **OK**.

Click **Next** to continue.

In the GS-Calc example, we expanded the selected files to confirm that the necessary components are selected. No filters have been applied.

Step 8: Setting File Properties and Locations

Use the **Instance Properties** dialog box to modify the properties of the files and folders in the package. The properties of a file or folder are stored in its instance in the Radia Database.

Use the **Radia Packager – Global Default Properties** dialog box to set the default values of these properties. For more information see *Setting Default Packaging Properties* on page 221.

To access the Instance Properties dialog box

1. Right-click the file or folder whose attributes you want to set.

2. Select **Set Properties**. If you selected a folder, you must also choose to select **Directory Only** or **Directory and Files**. The **Instance Properties** dialog box opens. It has four tabs: Client Management, Data Options, Client Behaviors, and Database Information.

Client Management Tab

Use the Client Management tab to set verification and delivery options for the selected file or folder in the package. After promoting this package to the Radia Database, you can still modify these settings using the Radia System Explorer.

■ Use the **verification options** to specify the verification actions that the Radia Software Manager will take for this file or folder.

■ Use the **delivery options** to specify delivery options, such as the order in which files are deployed.
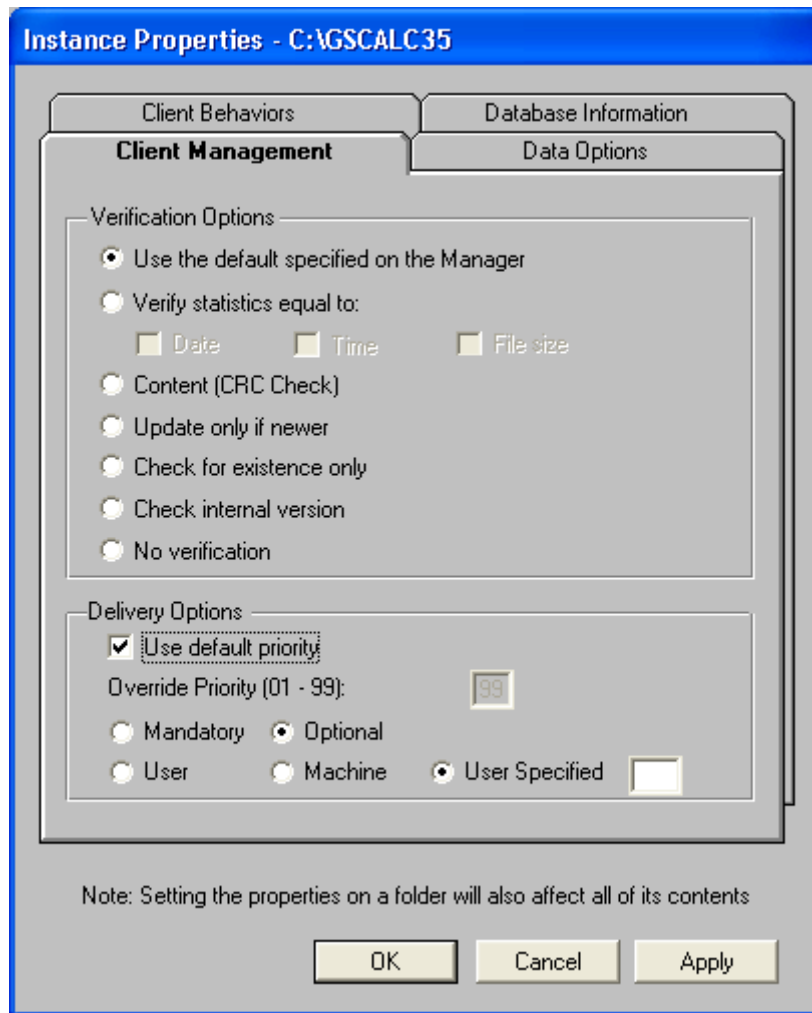
Figure 4.37 ~ Instance Properties dialog box, Client Management tab (Component Selection Mode).

VERIFICATION OPTIONS

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

Use the Radia System Explorer to look at the ZRSCVRFY attribute of the base instance of the FILE class to determine what the default verification options are.

■ **Verify statistics equal to**
Select this option so that the client checks the selected statistics (Date, Time, File Size) for the file on the client computer. The file is deployed from the Radia Database (or Radia Proxy Server) if its statistics are different from the file on the client computer. You can also use the Radia System Explorer to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.

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

■ **Update only if newer**
Select this option so that this file is deployed from the Radia Database (or Radia Proxy Server) if it has a later date/time stamp than the one on the client computer. You can also use the Radia System Explorer to set this option: ZRSCVRFY=U.

■ **Check for existence only**
Select this option so that this file is deployed only if it is not on the client computer. No action is taken if it already exists on the client computer, even if the statistics differ from those in the Radia Database. You can also use the Radia System Explorer to set this option: ZRSCVRFY=E.

■ **Check internal version**
Select this option to compare the version of the file being deployed against the version that may already exist on the client computer. This ensures that the correct file is being updated if more than one version exists. You can also use the Radia System Explorer to set this option: ZRSCVRFY=V. This feature applies to the following file types: DLL, .EXE, .OCX, .SYS, .VBX.

■ **No verification**
Select this option so that the file is deployed the first time the application is deployed. No subsequent action is taken. You can also use the Radia System Explorer to set this option: ZRSCVRFY=N.

DELIVERY OPTIONS

■ **Use default priority**
Select this check box to use the default priority of **50**. Priority determines the order deployment, from highest priority to lowest priority. You can also use Radia System Explorer to set this option: ZRSCPRI=50.

■ **Override Priority**
Type a number from **1** to **99** to override the default priority of **50**. **1** is the highest priority and **99** is the lowest. You can also use Radia System Explorer to set this option: ZRSCPRI=1.

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

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

● If there is not enough space on the client computer for the entire application, Radia will *only* deploy mandatory files.

● If there is not enough space for the mandatory files, then the application is not deployed at all.

You can also use the Radia System Explorer to set this option: ZRSCMO=M.

■ **Optional** (default)
Select this option to indicate that a file is not critical to the application.

● If there is not enough space on the client computer for the entire application, Radia will *not* deploy optional files.

You can also use the Radia System Explorer to set this option: ZRSCMO=O.

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

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

■ **Machine**
Select **Machine** to indicate that the file will be deployed to all users of the computer. You can also use the Radia System Explorer to set this option: ZCONTEXT=M.

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

In the GS-Calc example, we accepted the default verification and delivery settings.

Data Options Tab

Use the Data Options tab to specify data compression and other details about the files or folders that you will be distributing.
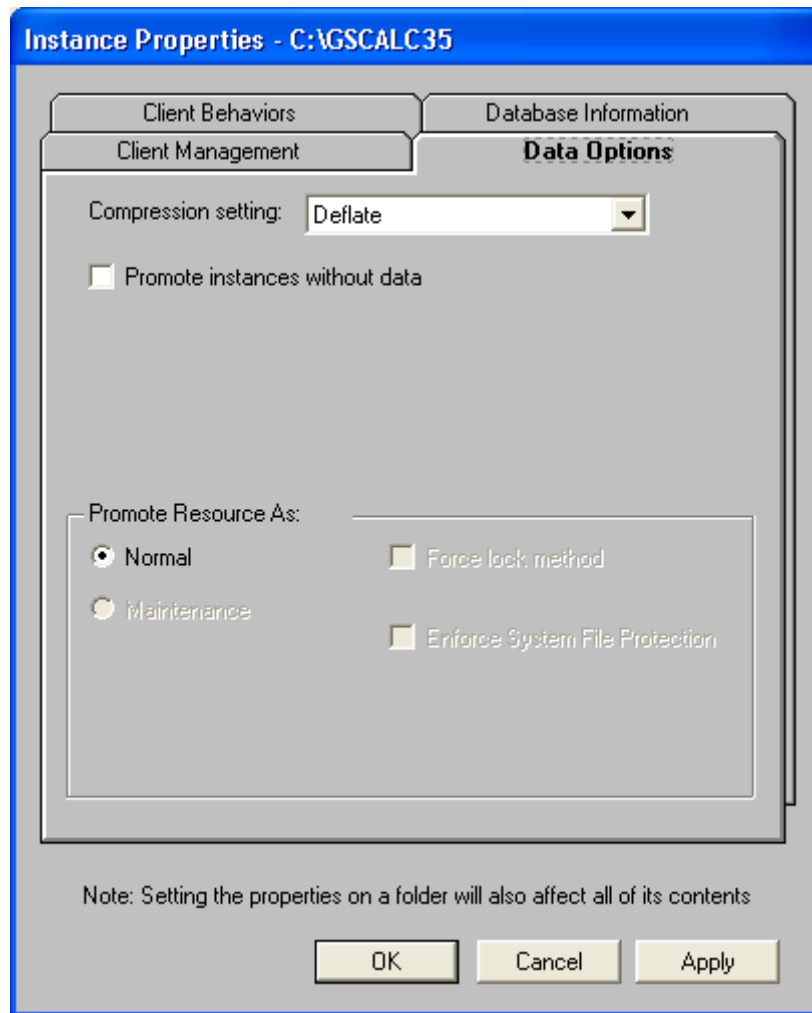
Figure 4.38 ~ Instance Properties dialog box, Data Options tab (Component Selection Mode).

■   **Compression setting**
    If necessary, select the compression setting for storing this file in the Radia Database.

Compression minimizes the time required to transmit it and the amount of disk space required to store it.

- Select **Deflate** for the most efficient compression, which produces smaller compressed images.

- Select **None** if the file is already compressed. If you are packaging an application that contains one or more compressed files, do *not* have Radia Packager compress the file as well. The file may actually grow in size if it is compressed again.

■ **Promote instances without data**
Select this check box to indicate that this file should not be transferred to the Radia Database as part of the package. Only the instance representing it is included in the package. The data remains in compressed format in the IDMDATA location on your computer. The default IDMDATA location is *SystemDrive***:\Program Files\Novadigm\Lib\Data**. You can manually place a file on a Radia Staging Server if you have connectivity to it and do not want a copy of it in the Radia Database.

PROMOTE RESOURCE AS

■   **Normal**
    Select this option to indicate that this file is to be deployed as part of an application.

■   **Maintenance**
    This is a feature from previous versions and, as of version 4.0, is no longer used.

■   **Force lock method**
    Select this check box to force the use of the *locked file method* for deploying this file. If the file
    is in use on the client computer when Radia attempts to deploy a new copy of it, the locked file
    method is normally used to deploy it. If necessary, this file is decompressed and stored locally
    in a folder. The Client Connect process forces a restart when it ends and the file is deployed to
    its correct location during the startup.

■   **Enforce System File Protection (Windows 2000 and Windows Server 2003 only)**
    Select this check box to specify that you want to enforce Windows System File Protection
    (SFP). Radia first attempts to verify the resource using the current verify options, as specified
    in the ZRSCVRFY attribute located in the FILE class.

    ●   If the verification is successful, the resource is in the desired state and no further
        verification is necessary.

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

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

    ●   If you select this check box, the ZRSCVRFY attribute is set to P and deployment of this
        resource and the service are marked as failed due to SFP processing.

    ●   If you do not enforce SFP, the file is marked as having passed verification.

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

In the GS-Calc example, we accepted the default data options.

Client Behaviors Tab

Use the Client Behaviors tab to specify methods (programs) that Radia executes on the client computer.
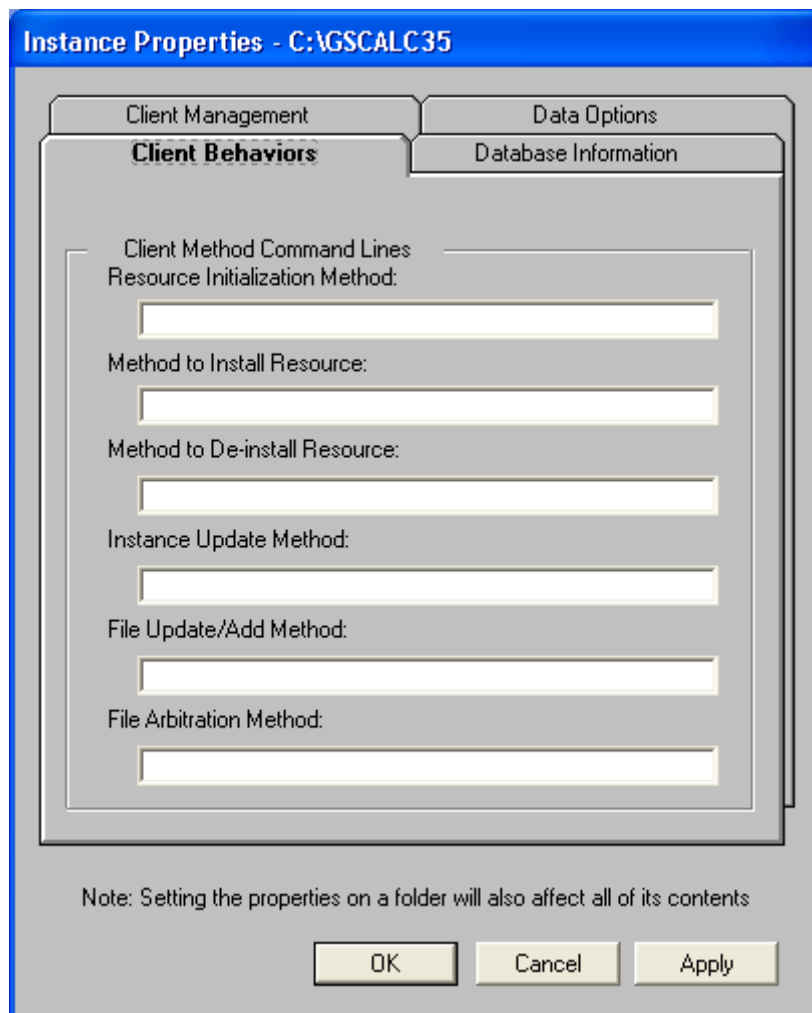
Figure 4.39 ~ Instance Properties dialog box, Client Behaviors tab (Component Selection Mode).

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

CLIENT METHOD COMMAND LINES

■   **Resource Initialization Method** (Attribute in database: ZINIT)
Type the method to run when the file is stored on the client computer.

■   **Method to Install Resource** (Attribute in database: ZCREATE)
Type the method to run after the file is stored on the client computer. This is used if some processing is required to enable it to be used on the client computer.

■   **Method to De-install Resource** (Attribute in database: ZDELETE)
Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared DLL file, is not supposed to be deleted from the client computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for the **Method to De-install Resource**.

■   **Instance Update Method** (Attribute in database: ZUPDATE)
Type the method to run when the instance is modified on the client computer after the file or folder has been deployed.

■   **File Update/Add Method** (Attribute in database: ZFILEUPD)
Type the method to run when the file is new or has been updated. The method executes just before the file is deployed to the client computer.

■   **File Arbitration Method** (Attribute in database: ARBITRAT)
Type the method to run if a file is about to be replaced. This method examines the version information of the file on the client computer and the one that is going to replace it, and then determines which to keep.

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

In the GS-Calc example, we have not specified any methods.

Database Information Tab

Use the Database Information tab to specify where the instance for the selected file or folder will be stored in the Radia Database.
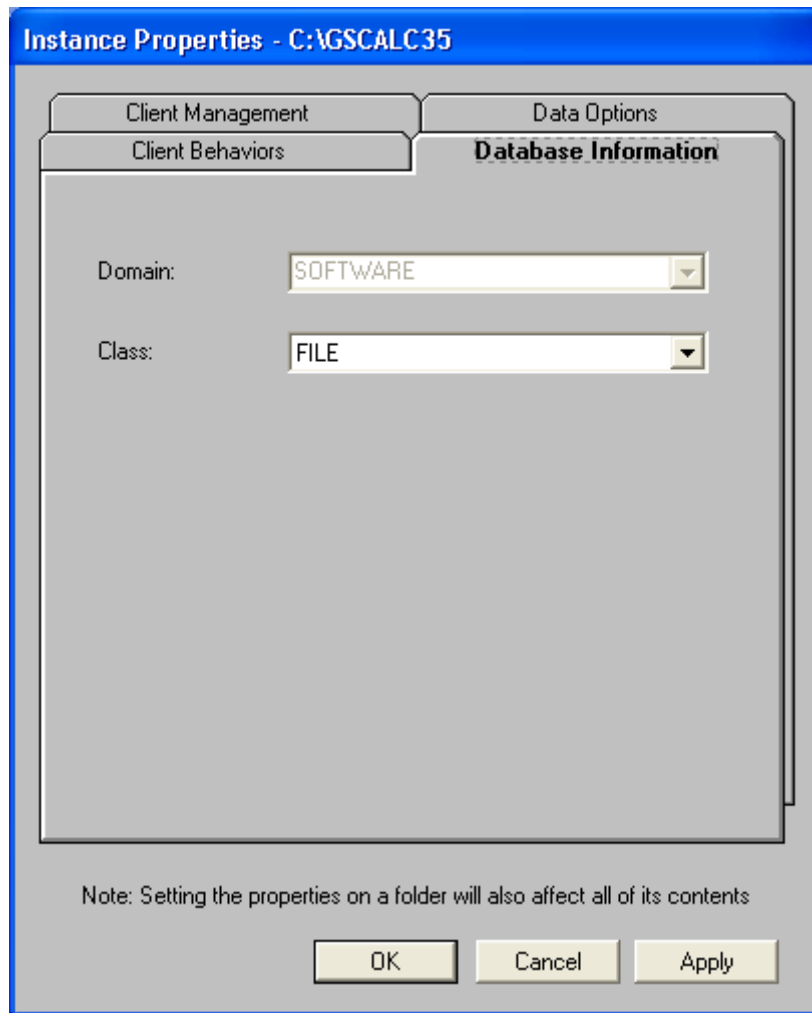
Figure 4.40 ~ Instance Properties dialog box, Database Information tab (Component Selection Mode).

■ **Domain**
    Select the domain in which to store the instance. This is normally the SOFTWARE domain unless you customized the Radia Database with proprietary domains.

- **Class**
  Select the component class in which to store the instance for this file.

In the GS-Calc example, we accepted the default settings for the domain and class.

Click **OK** to return to the **Set Properties and Locations** window, and then click the **Desktop** tab.

Step 9: Setting Desktop Properties and Locations

Use the **Desktop** tab in the **Set Properties and Locations** window to see the selected program groups, links, and shortcuts in the package, and to modify their properties. These program groups, links, and shortcuts will appear on the client computer.

To view the selected program groups, links, and shortcuts

■ Right-click **Selected Links** and select **Expand All**.
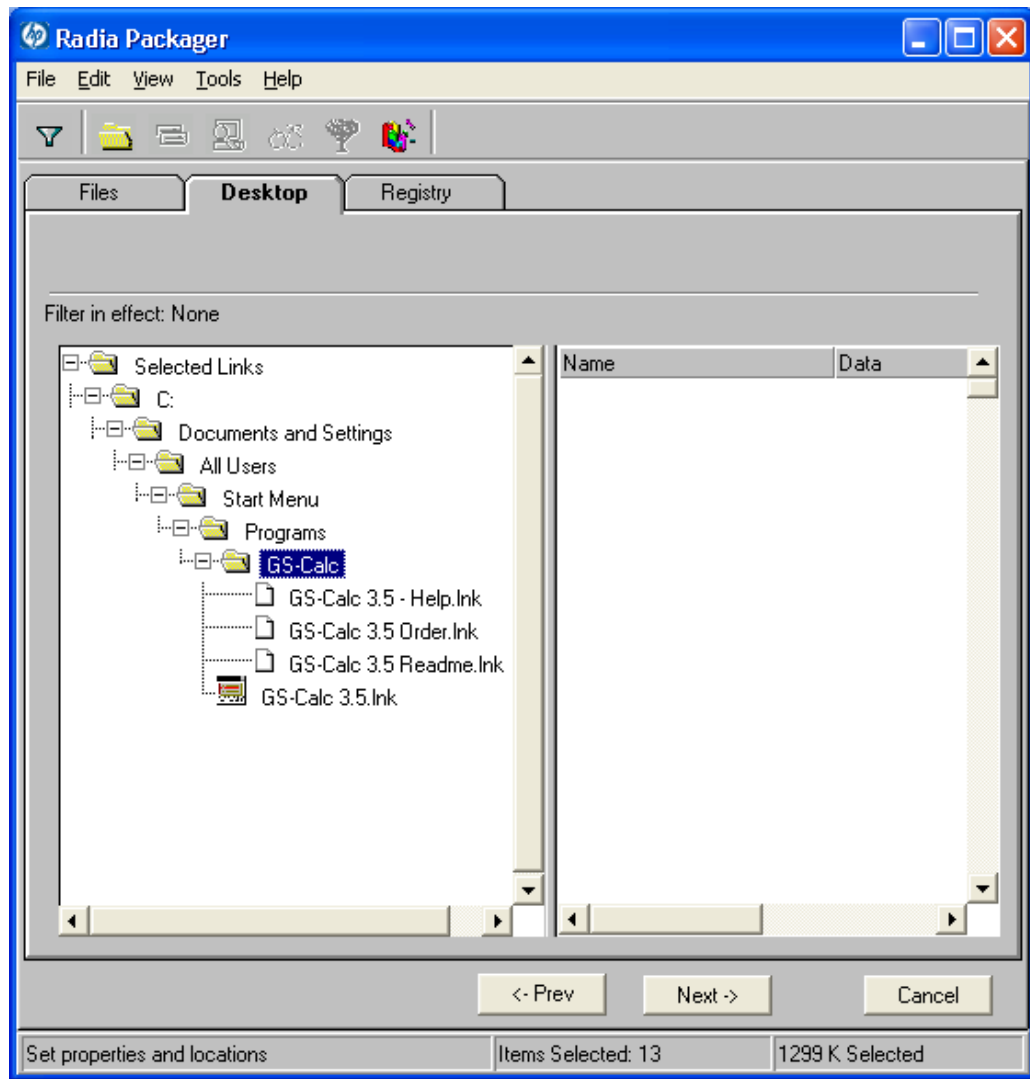
Figure 4.41 ~ View selected program groups, shortcuts and links (Component Selection Mode).

To set properties for program groups, links, and shortcuts

■   Right-click the link in the tree view. A menu opens.
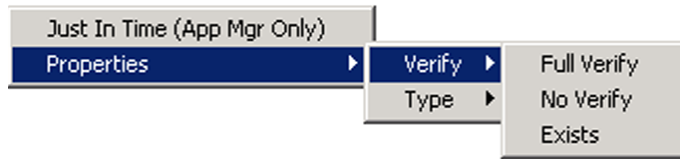
Figure 4.42 ~ Expanded shortcut menu (Component Selection Mode).

> **Note**
>
> When you right-click a folder or link, Radia applies the changes to all links within the folder or within the link's branch in the tree view.

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

■ **Properties**
Select **Properties** to open the **Verify/Type** submenu.

Verify Menu Commands

■ **Full Verify**
Select this command to deploy the link every time a subscriber connects to the Radia Configuration Server, even if it already exists on the client computer. This repairs broken or modified links.

■ **No Verify**
Select this command if you do *not* want to verify the link after initial deployment. No repairs are made to the link, even if the user modifies or removes it.

■ **Exists**
Select this command to deploy the link only if it does *not* exist on the client computer, during initial deployment and on each subsequent connection. The link is deployed again only if it was deleted. This allows users to modify the link.

Type Menu Commands

The following options apply only to operating systems supporting multiple users, such as Windows 95, 98, NT 4.0, 2000, or Windows Server 2003.

■ **User**
Select this command to deploy this link only to the subscriber logged on when the application is initially deployed.

■ **Machine**
Select this command to deploy this link to all users of the computer.

In the GS-Calc example, we left the default settings for program groups, links, and shortcuts. Click the **Registry** tab.

Step 10: Setting Registry Properties and Values

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

To view the selected registry

■ Right-click **Selected Registry Keys,** and select **Expand All**.
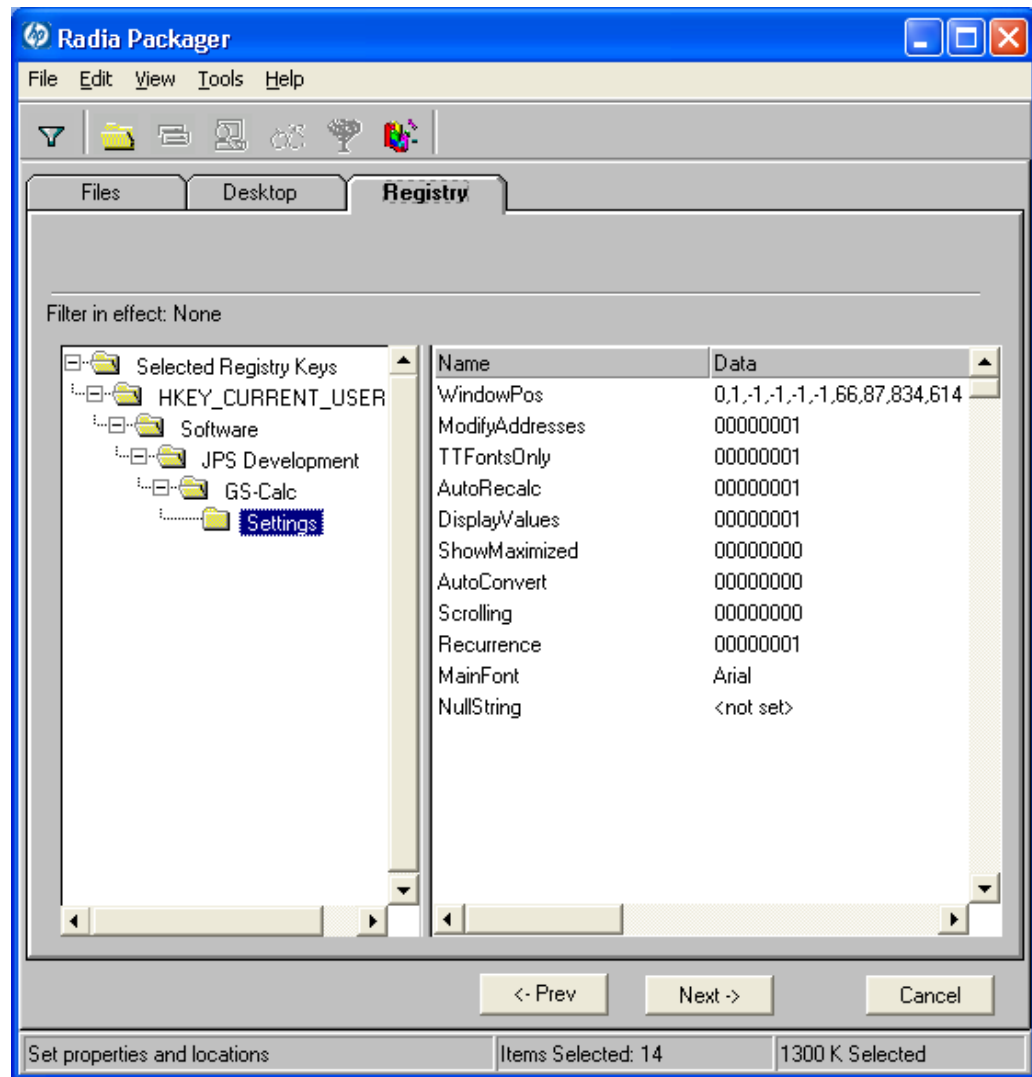
Figure 4.43 ~ View selected registry keys (Component Selection Mode).

To edit values in a registry key

1. Select the registry key in the tree view.

2. Double-click the registry key in the list view.

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

4. Click **OK** to save your changes and close the dialog box.
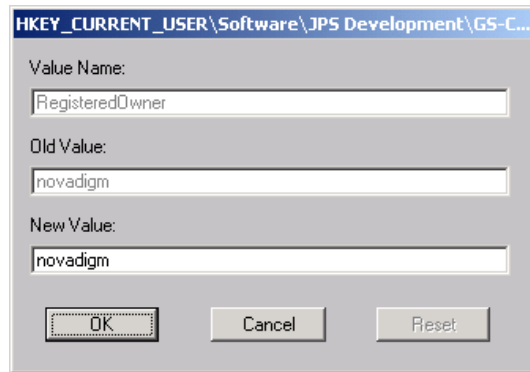
Figure 4.44 ~ Edit selected registry key (Component Selection Mode).

To restore the registry key to its original value

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

2. Click **Reset**.

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

To set properties for registry keys

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

Figure 4.45 ~ Expanded shortcut menu (Component Selection Mode).

> **Note**
>
> When you right-click a registry key, Radia applies the changes to all registry keys within the key's branch in the tree view.

- **Active**
  Select this command to include registry changes in the package.

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

- **Properties**

  - **Unconditional/Enforced**
    Select this command to deploy the registry key, even if it already exists on the client computer. Every time the subscriber connects to the Radia Configuration Server, this registry key is reset to the packaged value, if it has been modified on the client computer. Use this setting for registry keys that must not change in order for the application to function properly.

  - **Unconditional/Exist**
    Select this command to deploy the registry key upon initial deployment or if it has been deleted. However, the value of the registry key is not reset. This setting is the default.

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

In the GS-Calc example, the registry keys are set to **Unconditional/Exist**. Click **Next** to go to the **Promote Files** window.

Step 11: Promoting Packages

Use the **Promote Files** window to create instances for the package in the Radia Database.

To view the files to be promoted

■ Right-click **Files to be Promoted,** and select **Expand All**.
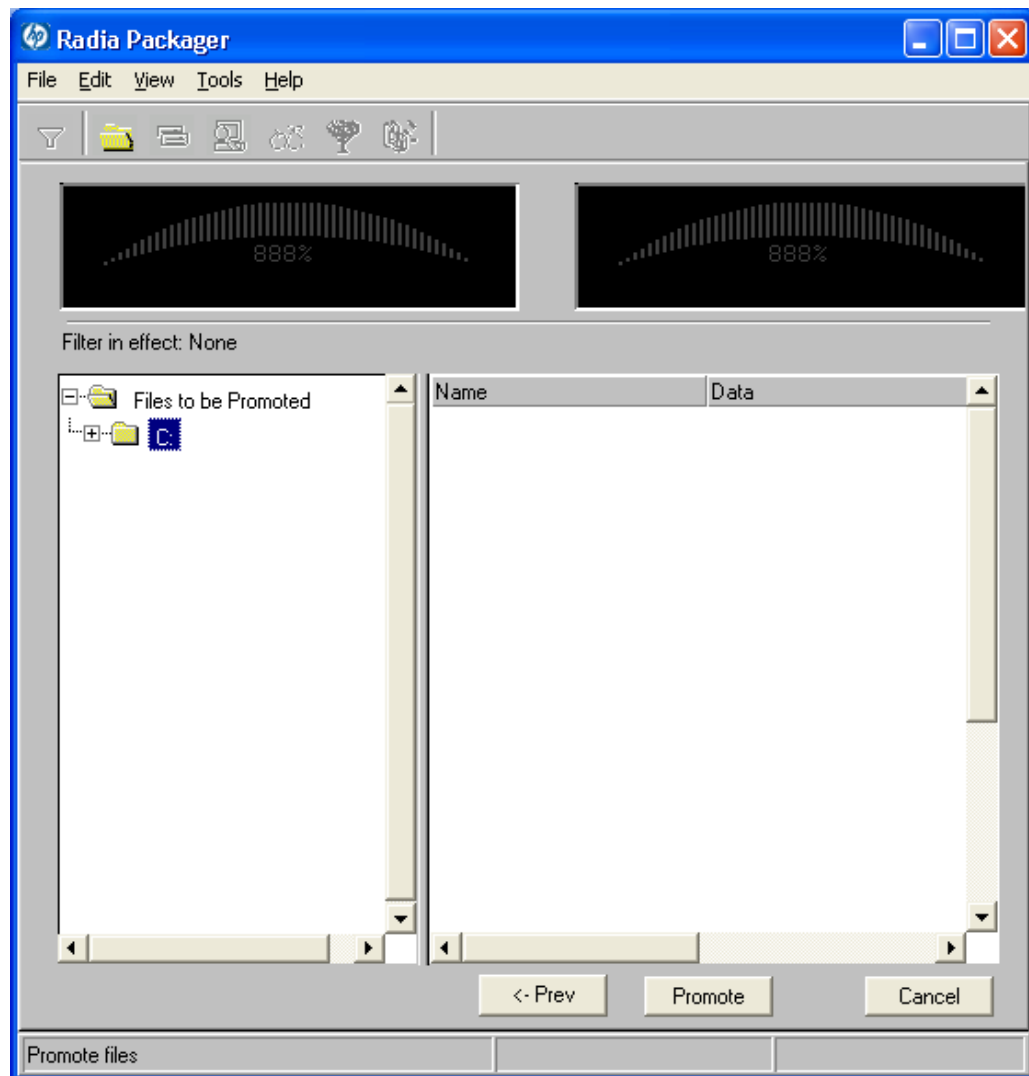
Figure 4.46 ~ Promote Files window (Component Selection Mode).

> **Note**
>
> If you need to change or modify your selections, click **Prev** until you reach the appropriate window. When you are satisfied with the package, click **Next** until you arrive back at this window.
>
> You can also use the buttons in the toolbar to return to a previous screen.

To promote the package

1. Click **Promote**.

2. Click **OK** when you receive the message that the package has been promoted successfully.

3. Click **Finish** to close Radia Packager.

4. Click **Yes** to confirm that you want to close the Radia Packager.

In this example, GS-Calc was promoted to the Radia Database. Next, use Radia System Explorer to create a service. For more information, see *Creating a Service* on page 212.

Radia Publishing Adapter

The Radia Publishing Adapter is a command-line alternative to using Component Selection Mode. Radia Publishing Adapter offers an automated, repeatable command-line process to create Radia packages and store them in the Radia Database for distribution.

Radia Publishing Adapter can:

■   Search for files on multiple drives/file systems.

■   Search for, and publish files from any mapped file/drive system.

■   Be configured to limit the subdirectories that are searched.

■   Include or exclude at the file level.

■   Select files by type.

Additionally, Radia Publishing Adapter can accommodate frequent patching of internal applications, as well as publish "build" versions, and output from legacy (PVCS or ClearCase) adapters. Its capacity to revise content material is reliable, and can be designed to perform continuously, at designated times, and in pre-determined intervals. RPA can be easily executed from within any script or code capable of calling a command prompt.

> **Note**
>
> Radia Publishing Adapter is an optional feature available from HP. Please contact your sales representative for more details.

Creating a Service

Once you have created a package with the Radia Packager, use the New Application Wizard in the Radia System Explorer to create a service. A *service* is the fundamental unit of content managed by Radia. Use the Radia System Explorer to see services listed in the Application (ZSERVICE) class.

In this example, you will create a service using the WinZip 8.0 package that you created in *Using Installation Monitor Mode*, beginning on page 123. Remember that the target platforms are Windows 95, 98, NT, 2000 and Windows Server 2003.

Using the New Application Wizard to Create a Service

Step 1: Access Radia System Explorer

To access the Radia System Explorer

1. Go to **Start**, **Programs**, **Radia Administrator Workstation**, and select **Radia System Explorer**.
2. In the **Radia System Explorer Security Information** dialog box, type your **User ID** and **Password** in the appropriate text boxes.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

3. Click **OK**.

Step 2: Navigate to the Package Class

To navigate to the PACKAGE class of the SOFTWARE domain

1. Double-click **PRIMARY**. The domains of the PRIMARY file appear beneath its icon in the tree view and in the list view.
2. Double-click **SOFTWARE**. The classes of the SOFTWARE domain appear beneath its icon in the tree view and in the list view.
3. Double-click **Application Packages (PACKAGE)** to open the PACKAGE class. The instances of the PACKAGE class appear beneath its icon in the tree view and in the list view.
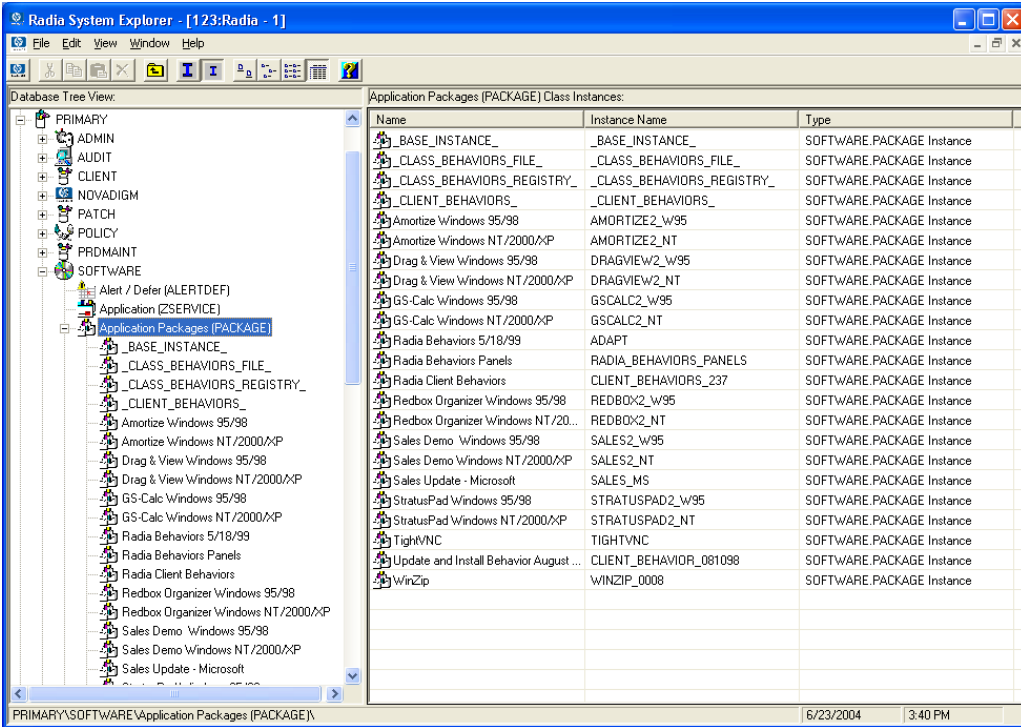
Figure 4.47 ~ Application Packages (PACKAGE) class.

Step 3: Create a Service

To create a service using the New Application Wizard

1. In the **PACKAGE** class of the **SOFTWARE** domain, right-click the **WinZip** instance. A menu opens.
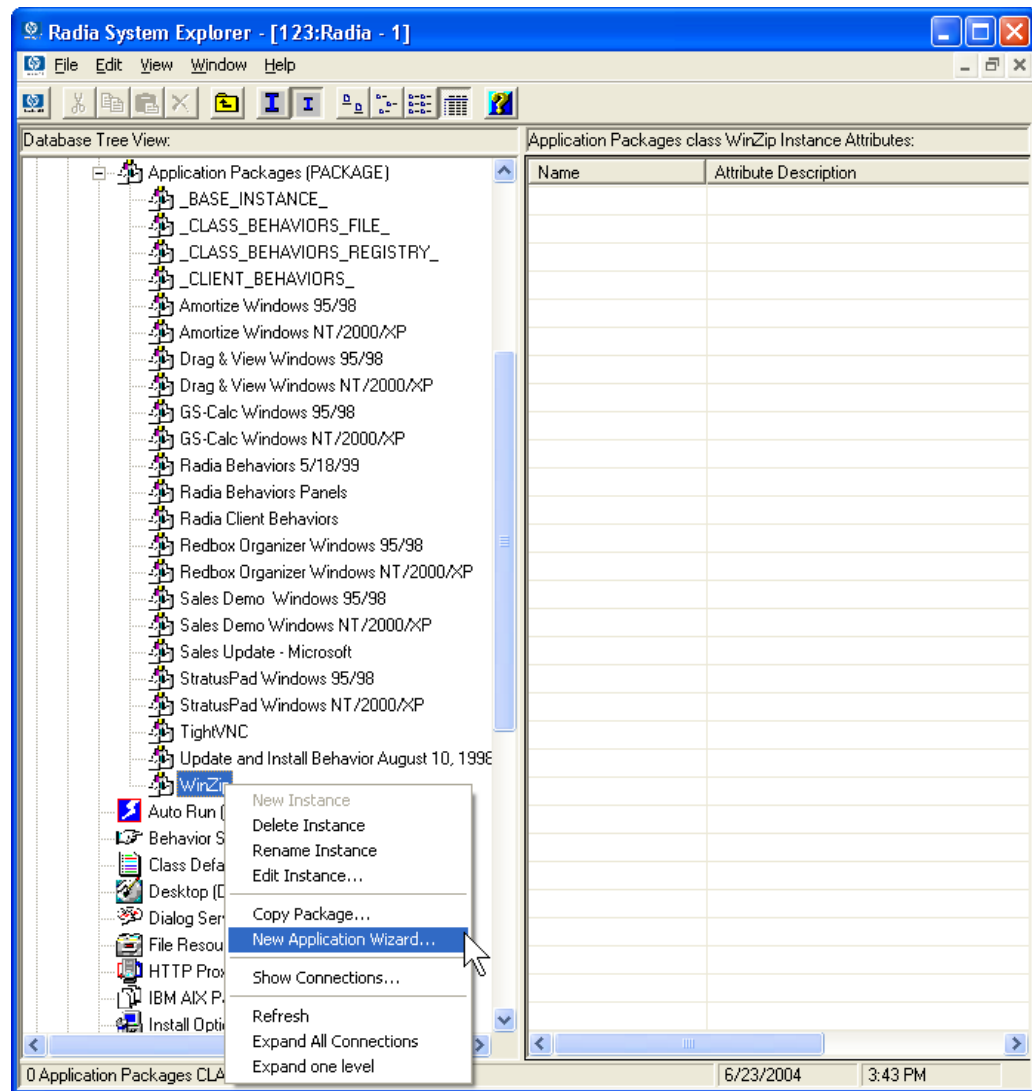
Figure 4.48 ~ Selecting New Application Wizard.

2. Click **New Application Wizard**.
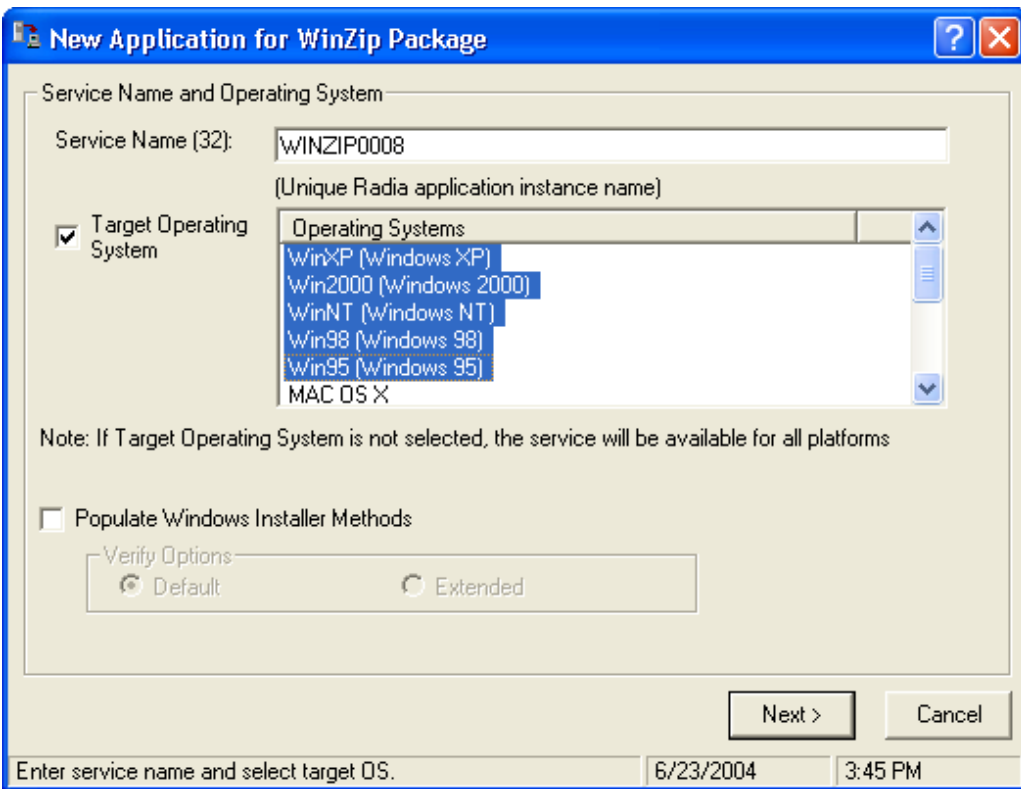
Figure 4.49 ~ Enter service name and select target OS dialog box.

3. In the **Service Name (32)** text box, type a name, such as **WinZip0008**, for the Application (ZSERVICE) instance.

4. Select the **Target Operating System** check box if you want to specify which operating systems are eligible for this service. Then, select the check boxes for the appropriate operating systems.

   If you do not select the **Target Operating System** check box, the service is available on *all* platforms.

5. If you are creating a service for a Windows Installer-enabled application, you must select the **Populate Windows Installer Methods** check box. *Do not select this check box for this exercise.*

6. Click **Next** to select the application target type.
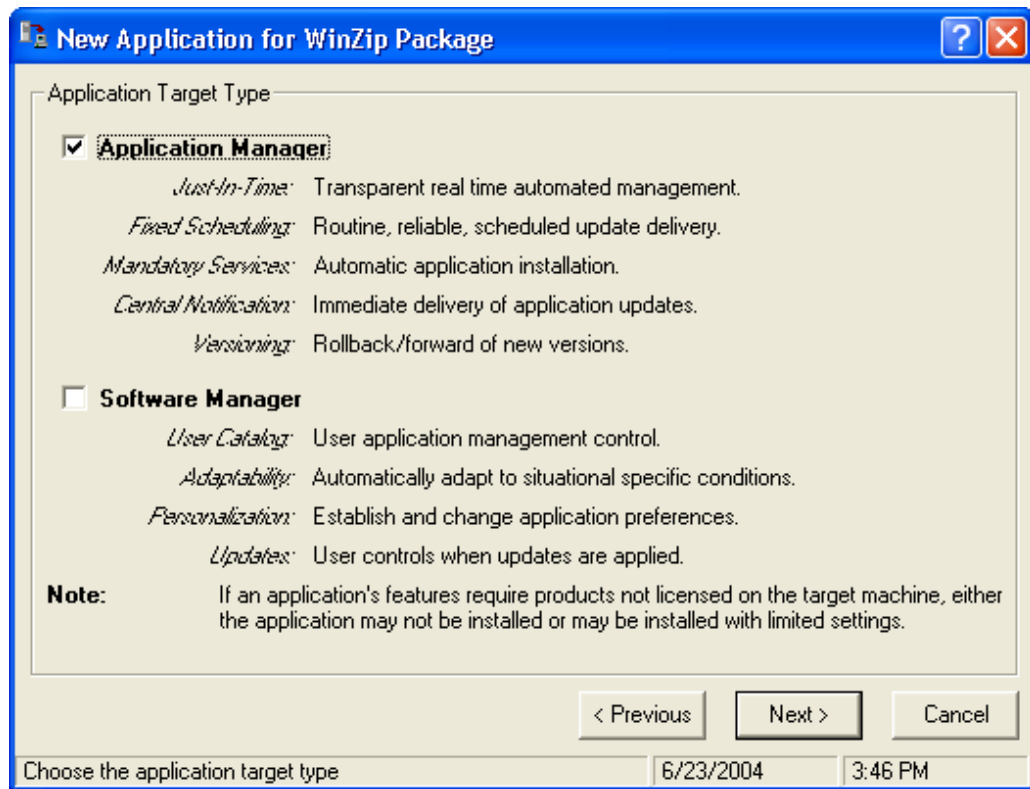
Figure 4.50 ~ Select the application target type dialog box.

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

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

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

Figure 4.51 ~ Enter the application properties.

9. Type the appropriate information in the text boxes as shown in Figure 4.51 above.

10. Click **Next** to select the events that the Radia Software Manager will report on.
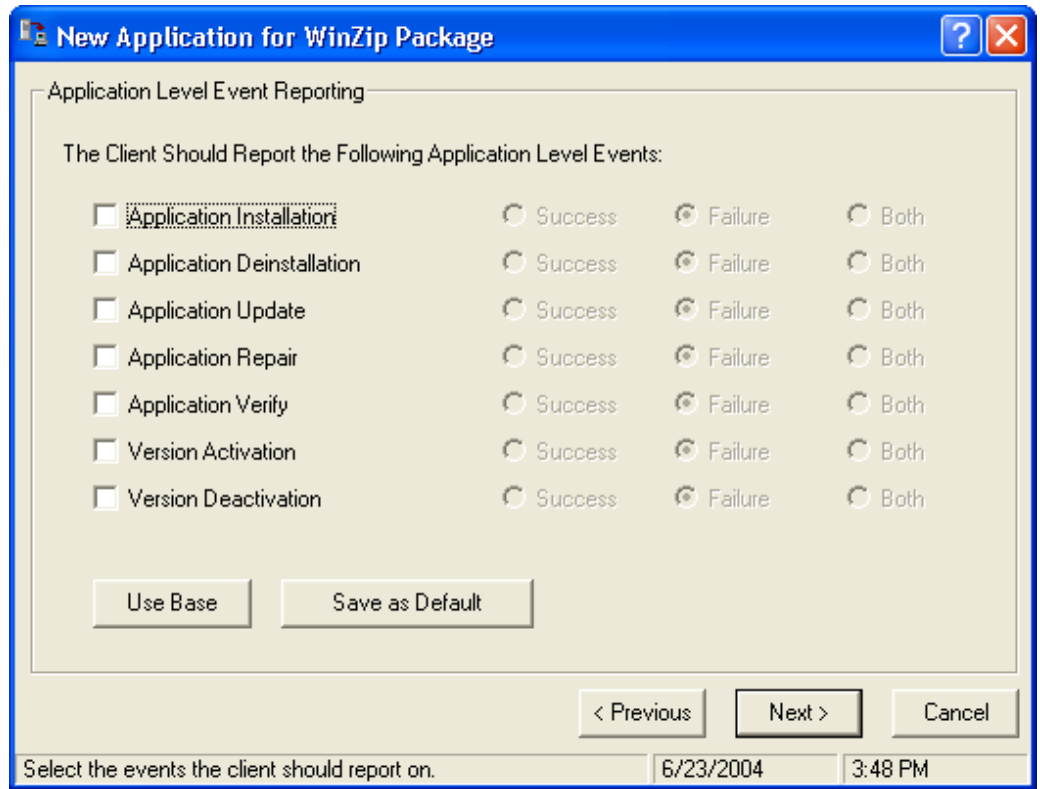
Figure 4.52 ~ Selecting the events that the Radia Software Manager will report on.

11. Click the check box for each event that you want to report on. Then, select the appropriate option button to indicate whether to report on the event's success, failure, or both.

OR

Click **Use Base** if you want to inherit the values for the ERTYPE and EVENTS attributes from the base instance of the Application (ZSERVICE) instance. These attributes control event reporting.

12. If you want to save the current settings as the default settings for the Application Event Panel, click **Save as Default**.

13. Click **Next** to review your selections.
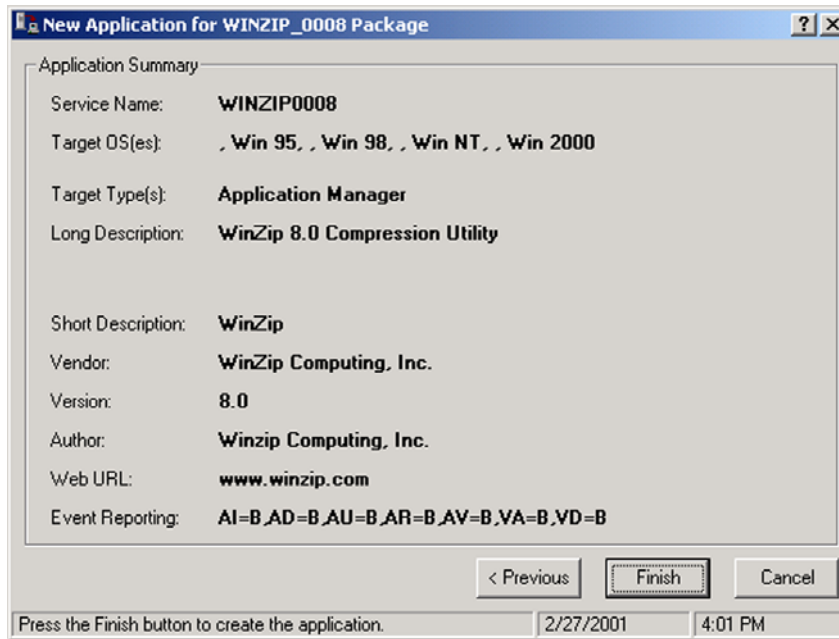
Figure 4.53 ~ Summary of the application settings.

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

15. Click **OK** when you are prompted with a message indicating that the application has been added. The instance appears in the ZSERVICE class.

> **Note**
>
> If you want to modify any of the information that you entered in the New Application Wizard, locate the corresponding attribute and change its value.
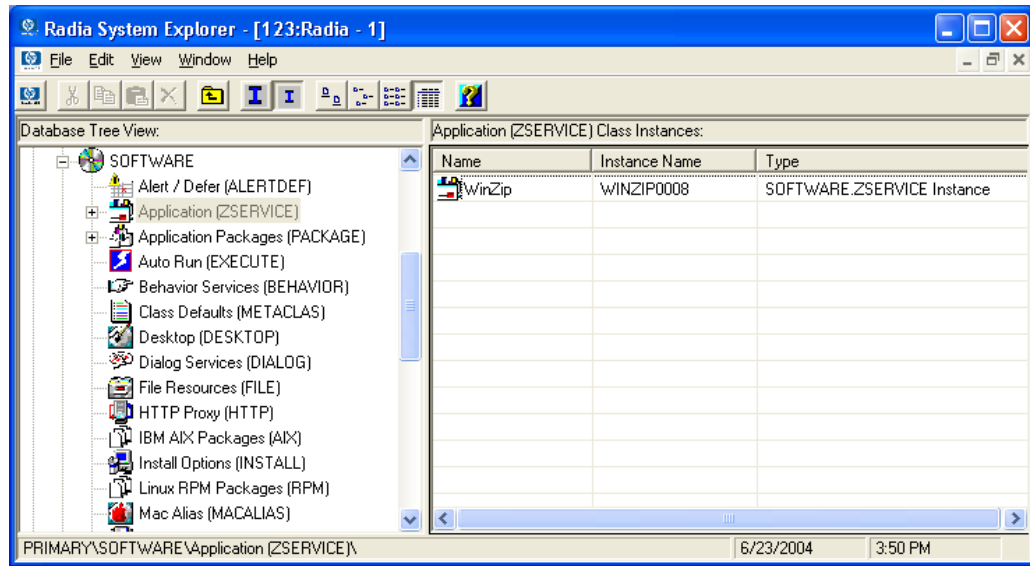
Figure 4.54 ~ New ZSERVICE instance.

Now, you are ready to set up policies identifying *which* subscriber receives *what* software. See *Chapter 5: Implementing Entitlement Policy* starting on page 233 for more information.

Setting Default Packaging Properties

You may want to set default properties (such as compression settings and verification options) for the files that will be included in the package. Changes made to the default properties apply to new Radia Packager sessions.

If necessary, you can modify the properties later for any file or folder from the **Set Properties and Locations** window. Learn how to do this in Installation Monitor Mode on page 150 and in Component Selection Mode on page 189.

You can also use the Radia System Explorer to modify the default properties in the base instance. Or, after you promote the package to the Radia Database, you can modify the properties for an individual instance in the FILE, REGISTRY, or DESKTOP classes.

To log on to Radia Packager

1. Go to **Start**, **Programs**, **Radia Administrator Workstation**, and select **Radia Packager**.

2. In the **Radia Packager Security Information** dialog box, type your **User ID** and **Password** in the appropriate text boxes.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

3. Click **OK**.

To access the Global Default Properties dialog box

■ From the **Edit** menu, select **Change Global Defaults**. The **Radia Packager – Global Default Properties** dialog box has three tabs: Client Management, Data Options, and Client Behaviors.

Client Management Tab

Use the **Client Management** tab to set verification and delivery options for the files in the package.
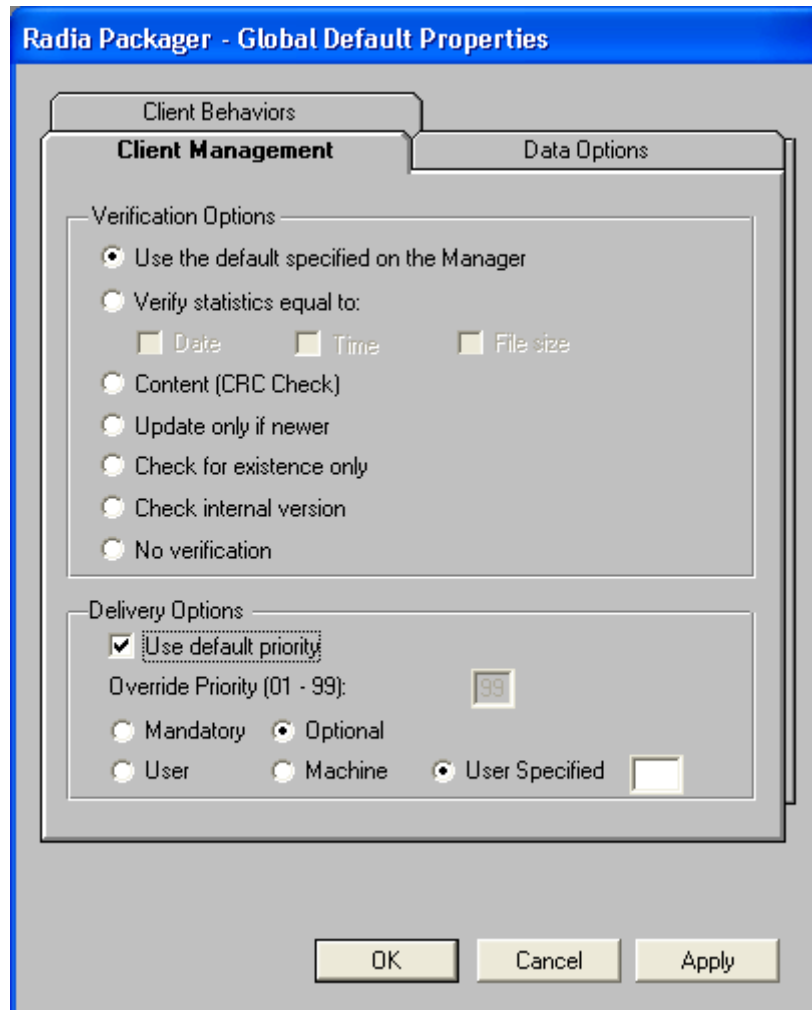
Figure 4.55 ~ Global Default Properties dialog box, Client Management tab.

Verification Options

Use **Verification Options** to specify the default actions that the Radia Software Manager will take for the files.

Table 4.1 ~ Verification Options		
Verification Option	**Usage**	**ZRSCVRFY Setting**
Use the default specified on the Manager	Select this option to inherit verification options from the FILE, REGISTRY, or DESKTOP class' base instance in the Radia Database.	ZRSCVRFY=<blank space>
Verify statistics equal to	Select this option so that the Radia Software Manager checks the selected statistics (Date [D], Time [T], File Size [S]) of the files on the client computer. The Radia Database (or Radia Proxy Server) will deploy a file if its statistics are different from those on the client computer.	ZRSCVRFY=D ZRSCVRFY=T ZRSCVRFY=S ZRSCVRFY=Y (to check date, time and size)
Content (CRC Check)	Select this option so the Radia Software Manager checks the statistics Content (CRC) of the file on the client computer. When the file is verified, it will *first* be checked based on the Date, Time, and File Size. To save time during the verify, the CRC will be checked *only* if the file passes the Date, Time, and File Size verification. Then, the Radia Configuration Server (or Radia Proxy Server) will deploy a file if its statistics are different from those on the client computer. By selecting this option, the Radia Packager will calculate the CRC and put the value in the ZRSCCRC attribute.	ZRSCVRFY=Y
Update only if newer	Select this option so a file is deployed from the Radia Database (or Radia Proxy Server) if is has a later date/time stamp than the one on the client computer.	ZRSCVRFY=U
Check for existence only	Select this option so a file is deployed only if it is not on the client computer. No action is taken if it already exists on the client computer, even if its statistics differ from those in the Radia Database.	ZRSCVRFY=E
Check internal version	Select this option to compare the version of the file being deployed against the version that may already be on the client computer. This ensures the correct file is being updated if more than one version exists. This applies to the following file types: DLL, .EXE, .OCX, .SYS, .VBX.	ZRSCVRFY =V
No verification	Select this option so the files are deployed the first time the software is deployed. No subsequent action is taken.	ZRSCVRFY=N

Delivery Options

Use **Delivery Options** to specify the default delivery options, such as the order in which files are deployed. Your selections in this dialog box set the corresponding attributes in the base instance of the FILE, REGISTRY, or DESKTOP classes. Use the Radia System Explorer to view or modify the appropriate attributes.

Table 4.2 ~ Delivery Options

Delivery Option	Usage	Attribute Settings
Use default priority	Select this check box to use the default priority of **50**. Priority determines the order of deployment, from highest priority to lowest priority.	ZRSCPRI=50
Override Priority	Type a number from **1** to **99** to override the default priority of **50**. 1 is the highest priority, 99 the lowest.	ZRSCPRI=1
The following options apply *only* if there is not enough space on the client computer to install the entire application.		
Mandatory	Select this option to indicate that the files are critical to the software. • If there is not enough space on the client computer for the entire application, Radia will only deploy mandatory files. • If there is not enough space for the mandatory files, then the software is not deployed at all.	ZRSCMO=M
Optional (default)	Select this option to indicate that the files are not critical to the software. If there is not enough space on the client computer for the entire application, Radia will not deploy optional files.	ZRSCMO=O
The following options apply only to operating systems supporting multiple users with a required sign on, such as Windows NT 4.0 or 2000.		
User	Select **User** to indicate that the files will be deployed only to the subscriber logged on when the application is initially deployed.	ZCONTEXT=U
Machine	Select **Machine** to indicate that the files will be deployed to all users of the computer.	ZCONTEXT=M
User Specified	*This option is for future use.*	*For future use.*

Data Options Tab

Use the **Data Options** tab to specify the default data compression and other details about the files you will be distributing.
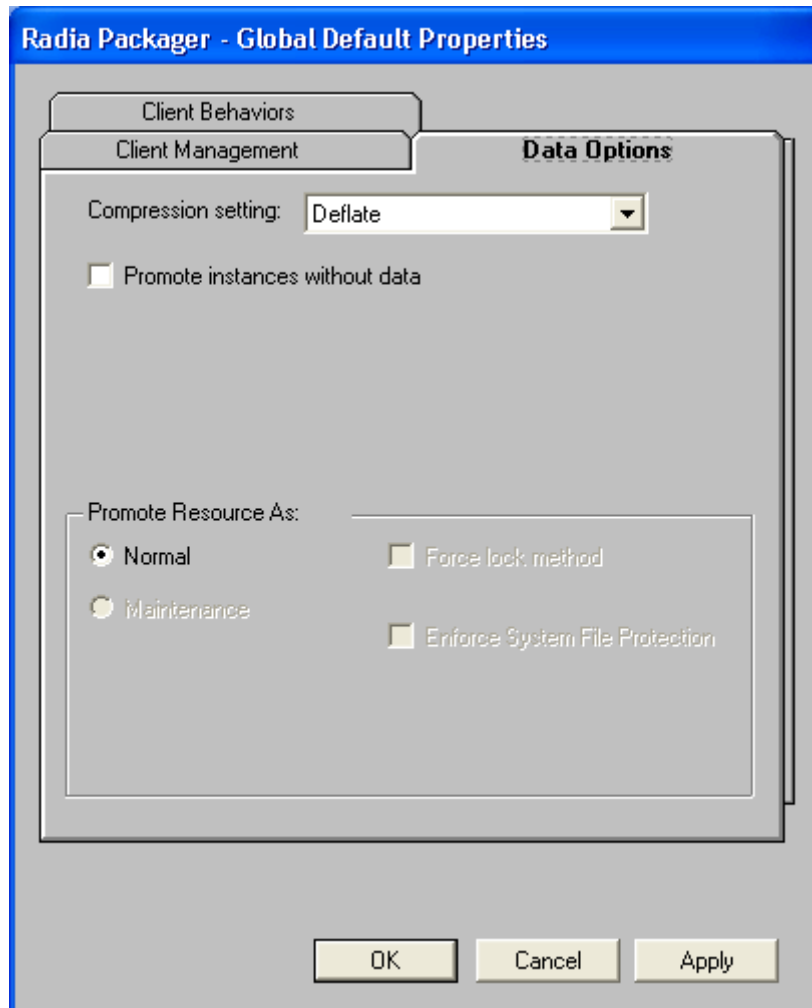
Figure 4.56 ~ Global Default Properties dialog box, Data Options tab.

Table 4.3 ~ Data Options

Data Option	Usage
Compression setting	If necessary, select the compression setting for storing files in the Radia Database. Compression minimizes the time required to transmit the files and the amount of disk space required to store them. • Select **Deflate** for the most efficient compression, which produces smaller compressed images. • Select **None** if the files are already compressed. If you are packaging an application that contains one or more compressed files, do *not* have the Radia Packager compress them as well. The files may actually grow in size if they are compressed again.
Promote instances without data	Select this check box to indicate that the files should not be transferred to the Radia Database as part of the package. Only the instances representing the files are included in the package. The data remains in compressed format in the IDMDATA location on your computer. The default IDMDATA location is *SystemDrive*:**\Program Files\Novadigm\Lib\Data**. You can manually place files on a Radia Staging Server if you have connectivity to it and do not want copies of them in the Radia Database.

Table 4.3 ~ Data Options

Data Option	Usage
Promote Resource As	
• **Normal**	Select this option to indicate that the files are to be deployed as part of an application. This is the default selection.
• **Maintenance**	This is a feature from previous versions and, as of version 4.0, is no longer used.
• **Force lock method**	Select this check box to force the use of the *locked file method* for deploying the files. If a file is in use on the client computer when Radia attempts to deploy a new copy of it, the locked file method is normally used to deploy it. If necessary, the file is decompressed and stored locally in a folder. The Client Connect process forces a restart when it ends and the file is deployed to its correct location during startup.
• **Enforce System File Protection** (*Windows 2000 only*)	Select this check box to specify that you want to enforce Windows System File Protection (SFP). Radia first attempts to verify the resource using the current verify options, as specified in the ZRSCVRFY attribute located in the FILE class. • If the verification is successful, the resource is in the desired state and no further verification is necessary. • If the verification fails, Radia checks to see if System File Protection protects the file. If the file is protected, it will not be deployed. Finally, Radia checks this attribute to see whether you want to enforce SFP. • If you select this check box, the ZRSCVRFY attribute is set to P and deployment of this resource and the service are marked as failed due to SFP processing. • If you do not enforce SFP, the file is marked as having passed verification. A ZERROR instance is created to describe the actions taken against the SFP file.

Client Behaviors Tab

Use the **Client Behaviors** tab to specify default methods (or programs) that Radia executes on the client computer.
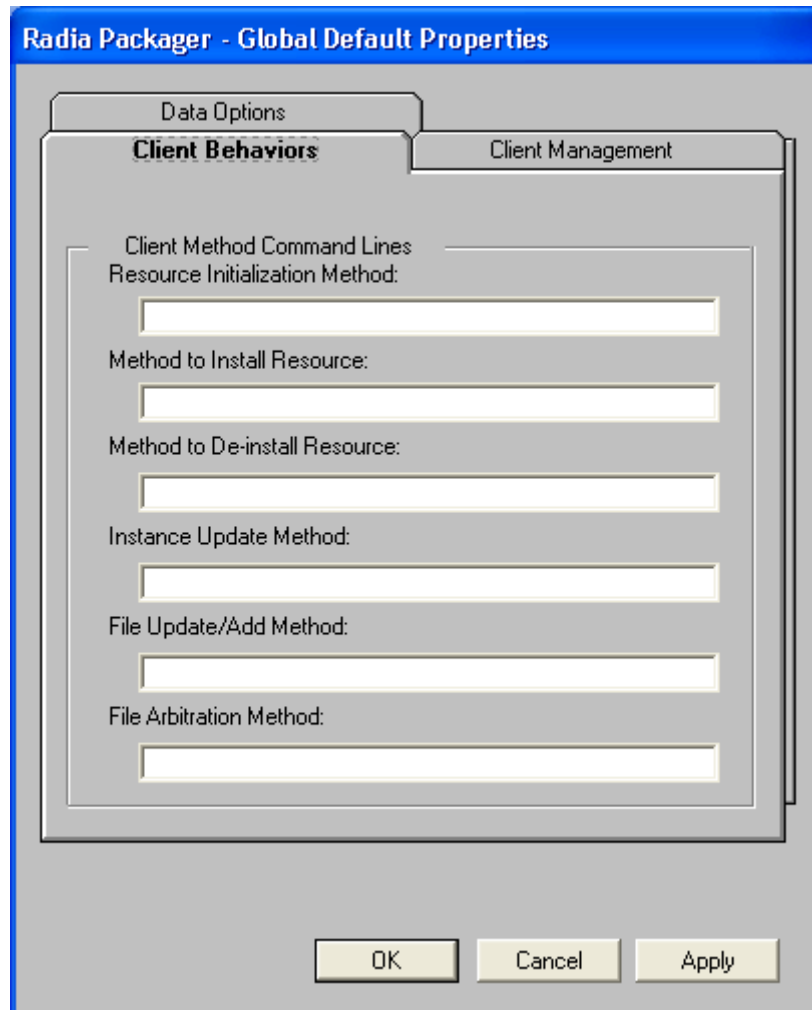
Figure 4.57 ~ Global Default Properties dialog box, Client Behaviors tab.

The command lines that you type in this dialog box are stored in attributes in the FILE class instances in the SOFTWARE domain. The following table describes the fields available on the

Client Behaviors tab, as shown in *Figure 4.57 ~ Global Default Properties dialog box, Client Behaviors tab* on page 228.

Table 4.4 ~ Client Behaviors	
Client Behaviors	**Usage**
Resource Initialization Method (Attribute in database: ZINIT)	Type the method to run when the files are stored on the client computer.
Method to Install Resource (Attribute in database: ZCREATE)	Type the method to run after the files are stored on the client computer. This is used if some processing is required to enable them to be used on the client computer.
Method to De-install Resource (Attribute in database: ZDELETE)	Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared DLL file, is not supposed to be deleted from the client computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for Method to De-install Resource.
Instance Update Method (Attribute in database: ZUPDATE)	Type the method to run when an instance is modified on the client computer after the file has been deployed.
File Update/Add Method (Attribute in database: ZFILEUPD)	Type the method to run when a file is new or has been updated. The method executes just before the file is deployed to the client computer.
File Arbitration Method (Attribute in database: ARBITRAT)	Type the method to run if a file is about to be replaced. This method compares the version information of the file that exists and the one that is going to replace it, and then determines which one to keep. **Note:** You must add the ARBITRAT method to the Radia Database to use this text box. See the Technical Note about the File Arbitration Method on the HP OpenView web site for more information.

Radia Service Groups

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

This includes applications installed or maintained using Windows Installer where:

■   A product may utilize more than one MSI service-package, for example, where MS Office may require other language pack services.

■   A large product, MS Office for example, may need to be split into smaller sub-services to install only specific parts of the product suite.

The Radia Software Manager user interface will display only the master service.

For detailed information on creating Radia Service Groups, refer to the *Radia System Explorer Guide.*

Optimizing Services

Service Optimization uses byte level differencing and its ability to generate *patches* to recreate original data. A *patch* allows administrators to upgrade data to reflect bug fixes, feature additions, and added information. These patches contain the minimum number of bytes required to fix a flawed program and/or complete software upgrades. These "patches" are smaller than the data, thus conserving network bandwidth at the expense of CPU overhead.

The Radia Packager automatically creates components that are eligible for byte-level differencing patching assuming that the component class contains the proper signature attributes as specified in the Radia Configuration Server specifications.

Note
▪   Patches can only be created for components that contain a signature. Initially, only MD5 is supported.
▪   Components being used for patching must be packaged 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.

For detailed information, see the *Radia System Explorer Guide.*

Windows System File Protection

Radia enforces System File Protection (SFP) on Windows Installer, and other packaged content, defers deployment of SFP-protected files, and reports SFP conflicts to administrators.

Radia will first attempt to verify the resource using the current verify options, as specified in the ZRSCVRFY attribute located in the FILE class.

■ If the verification is successful, the resource is in the desired state and no further verification is necessary.

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

Finally, Radia checks to see whether you want to enforce SFP. To enforce SFP, use Radia Packager and select the **Enforce System File Protection** check box on the Data Options tab in the **Instance Properties** dialog box for the selected file. See Table 4.3 on page 226 for a complete description of System File Protection. If you select this check box, the ZRSCVRFY attribute is set to **P**.

■ If you enforce SFP, deployment of this resource and the service are marked as failed due to SFP processing.

■ If you do not enforce SFP, the file is marked as having passed verification.

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

For more details on System File Protection, see the Microsoft Web site.

Summary

■ Packaging is the process of identifying the components of the software or content and organizing them into packages.

■ To package and configure applications, install the Radia Administrator Workstation on a clean computer. You can install these tools on the same computer as the Radia Configuration Server if you are running Windows NT 4.0 (or greater), or Windows 2000 or Windows Server 2003.

■ You can use Installation Monitor Mode to package applications when you do not know all of the components that make up the application. Radia scans the computer before and after you install the application and differences these scans to make up the package.

■ You can use Component Selection Mode for packaging simple applications by selecting the individual components that make up the software.

■ Use the Radia Publishing Adapter as an alternative to Component Selection Mode.

■ After packaging applications, use the New Application Wizard in the Radia System Explorer to create a service—the fundamental unit of content managed by Radia.

5

Implementing Entitlement Policy

- ■ Understand how Radia can integrate with your existing policy information.
- ■ Understand the Radia POLICY domain.
- ■ Be able to create new users and assign them to groups for use in simple environments.
- ■ Be able to connect services to groups.

This guide covers the *standard* implementation for the Radia Software Manager. Although you will tailor this strategy to meet your organization's needs, it is recommended that you review this guide for a comprehensive understanding of the Radia Software Manager. This chapter introduces you to how you can integrate your existing policy information with Radia. It also covers the POLICY domain, and provides background information about how to create users and groups, assign users to groups, and connect services to groups.
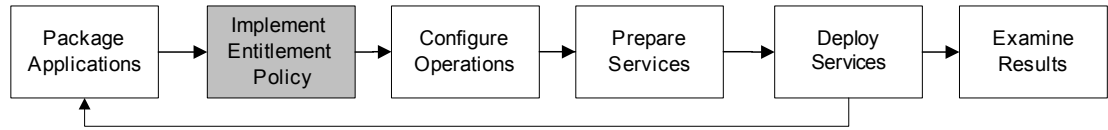
Figure 5.1 ~ Overview of the Radia Software Manager Guide.

About Policy Management and Radia

As your organization grows and changes, it is your job to manage *who* has access to *what* software. You've invested time and money to determine the best way to handle policy information for your organization. Now, you want to use Radia to manage your digital assets. With Radia, you have the advantage of using your *existing* policy information, while using Radia to manage your digital assets.

Radia can use real-time policy information from:

- Active Directory
- NDS
- iPlanet
- ISOCOR
- SQL Server, Oracle, or Sybase
- SQL 92-compliant (ODBC) data sources
- Any LDAP-compliant directory

You can continue to use the tools that you are already familiar with to administer policies. And, as you modify group assignments, subscriptions to digital assets are kept up-to-date.

Accessing Existing External Policy Information

When a Radia Client connects to the Radia Configuration Server, Radia retrieves policy information in real-time from the appropriate data stores. In the simplest environment, such as a lab used for testing, you might want Radia to search the Radia Database for this information. In a large-scale environment where an external policy store already exists, Radia can leverage this existing information. This information is sent back to the Radia Configuration Server, which determines which digital assets are to be managed for the user, group, or computer.

> **Note**
>
> For technical details about integrating your existing policy using the Radia Policy Server, see the HP OpenView web site and the *Radia Policy Server Guide*.
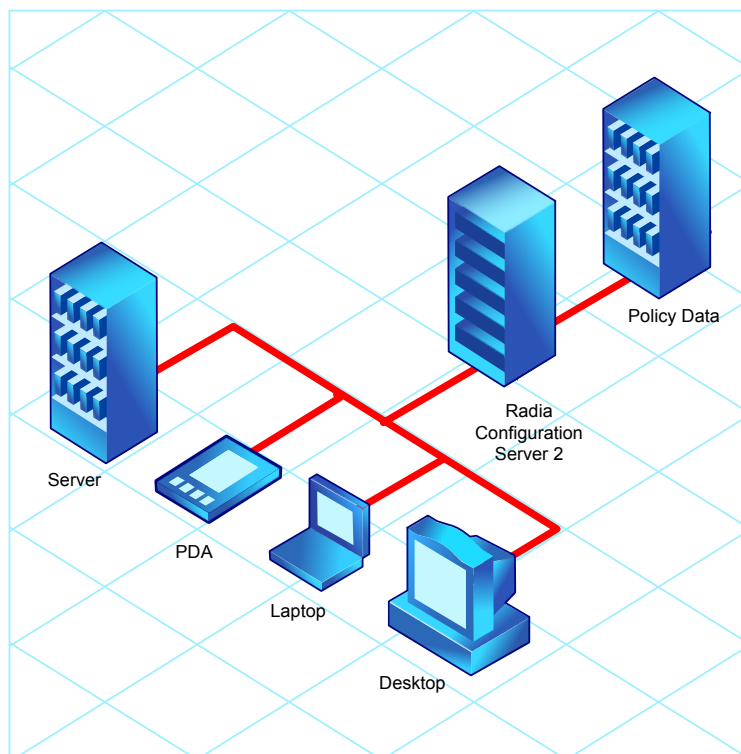
Figure 5.2 ~ Retrieving policy information from an external source.

Radia also supports using multiple Radia Configuration Servers with multiple types of external policy stores. This is especially useful in migration scenarios where you may be consolidating multiple external policy stores over a period. During this time, you can continue to use as many existing policy stores as necessary.
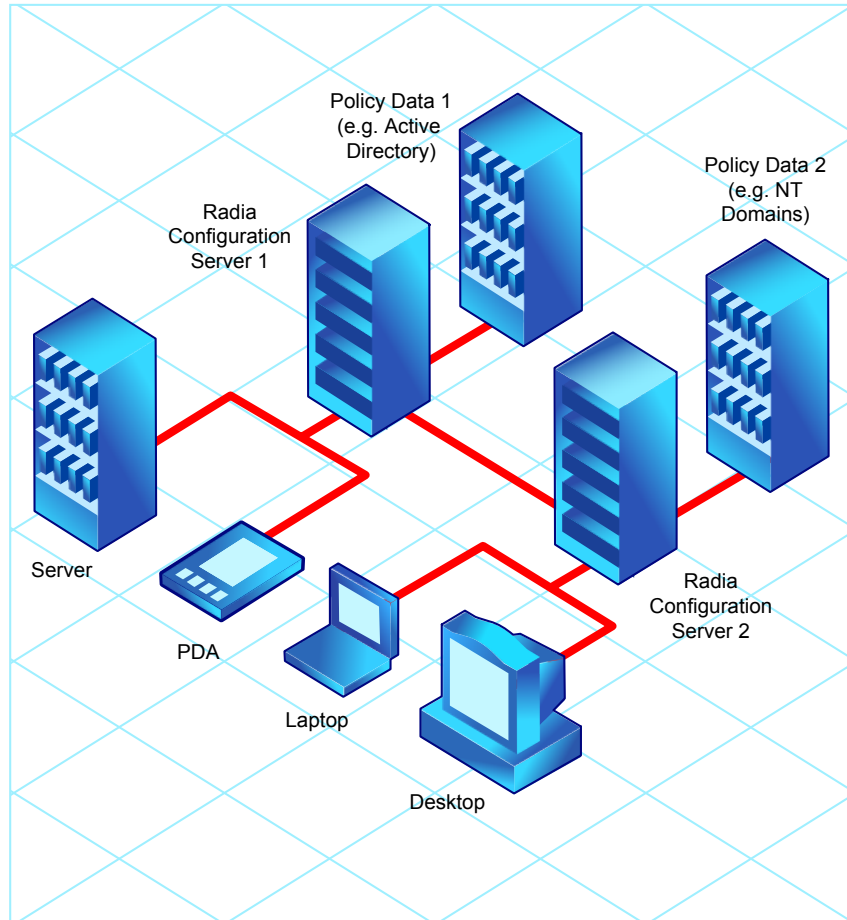
Figure 5.3 ~ Retrieving policy information from multiple external sources.

Directories-Based Entitlement

(such as Active Directory and NDS)

If you want to leverage your investment in LDAP-based directory services or SQL-based databases, HP offers the Radia Policy Server. The *Radia Policy Server* is a plug-in to the Radia Integration Server (RIS) used for administration purposes such as mapping services to users or computers in the directory tree. The Radia Configuration Server can be configured to query the Radia Policy Server to determine what services should be distributed and managed for the client.

> **Important Note**
>
> The Radia Policy Server is an optional feature available from HP. Contact your HP sales representative for details.
>
> See the *Radia Policy Server* document on our HP OpenView web site for more information.

Radia's integration with existing policy greatly reduces the total cost of ownership of your environment by allowing you to continue to manage policies from your existing repository while Radia manages your digital assets.

About the Radia POLICY Domain

If you are using real-time policy information from an external source, such as NT domains, to manage your digital assets, you may need to configure a connection from your external policy store to the POLICY domain in the Radia Database. The configuration may vary based on the policy store.

This section is intended to provide you with an overview of the POLICY domain. Most medium to large organizations will use its existing policy information and will have limited use for this domain. However, in the simplest environment, you can use the POLICY domain in the Radia Database to organize subscribers into logical groups in preparation for distributing software.

In this section, you will learn:

■ About the classes in the POLICY domain.

■ How to create users and groups.

■ How to assign users to groups.

Once you are familiar with the POLICY domain and understand the basics of managing policy information within Radia, you can extend that knowledge to learn how to integrate your existing policy information with Radia. This information may also be useful if you want to create a simple lab environment to test the management of your digital assets.

To access the POLICY domain

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

Note
The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

   The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **POLICY**.
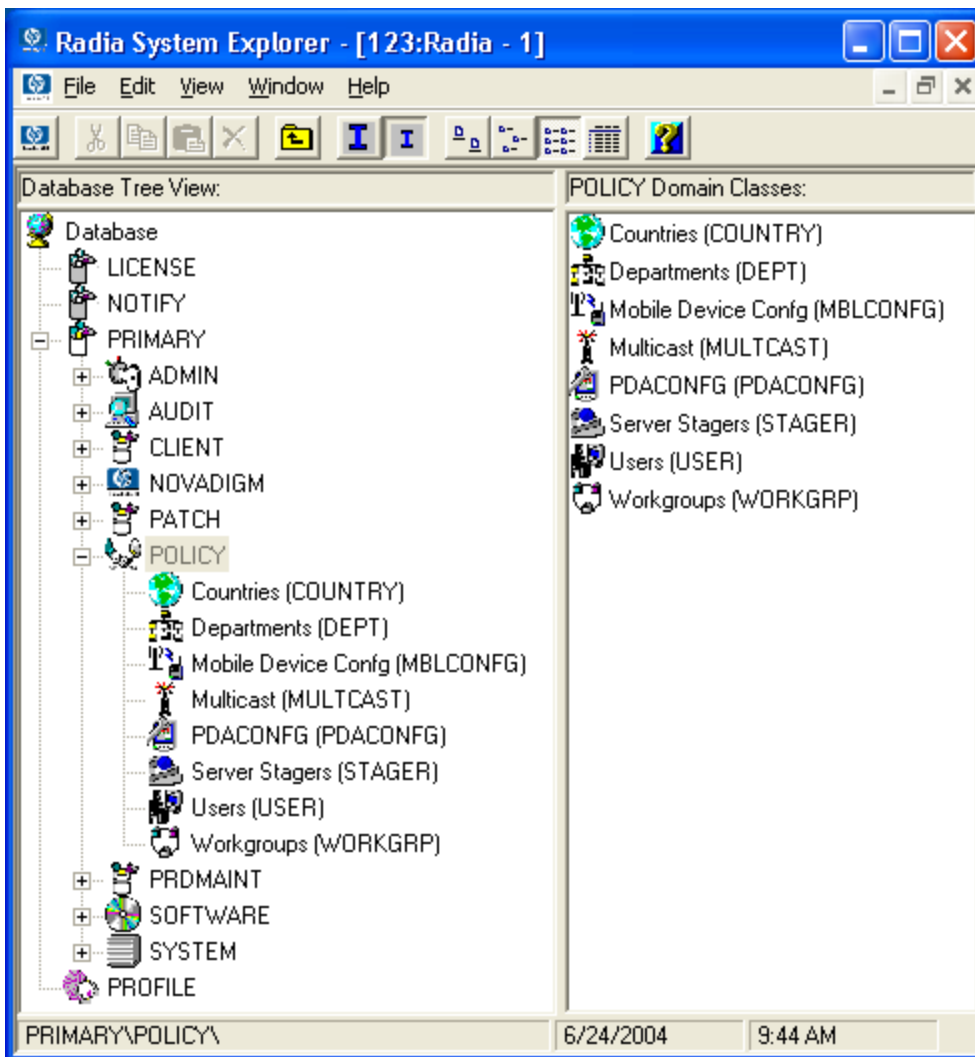
Figure 5.4 ~ The POLICY domain.

Classes in the POLICY Domain

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

Table 5.1 ~ Classes in the POLICY Domain		
Class	**Description**	**Instance Examples**
Countries (COUNTRY)	Use for clock synchronizations with the Radia Configuration Server. Do not assign services to this class.	France, Japan, Italy
Departments (DEPT)	Use to group subscribers into departments.	Finance, Customer Service, Manufacturing
Mobile Device Config (MBLCONFG)	Defines the parameters for mobile device configuration when using the Radia Mobility Server.	RmmUser
Multicast (MULTCAST)	Use the MULTCAST class to configure client computers to use multicasting.	MCast1, Mcast2
PDACONFG (PDACONFG)	This class defines the parameters for PDA configuration.	PDAUser
Server Stagers (STAGER)	Use to define Radia Proxy Servers or Radia Staging Servers within your distribution network.	CDROM, RPS, RPSUser
Users (USER)	Use to define individual subscribers. This can be either a user name *or* a computer name.	William, John Doe, SSampson
Workgroups (WORKGRP)	Use to group subscribers into functional groups. For example, a project team may be made up of subscribers from several different departments.	Project Planning, Managers, ABC Project Team

You can also add other classes to the POLICY domain, as per your organization's needs. For example, if your organization is an insurance company, you may add an AGENTS or OFFICES class. If your organization is a bank, you might add classes such as BRANCHES or TELLERS to organize your subscribers.

Note
See the *Radia System Explorer Guide* for information about creating new classes.

Creating Users or Groups in Radia

There may be times when you need to create individual users or groups in Radia. For example, you might want to create a lab environment used to test the distribution and management of your digital assets. To create a simple environment, you may want to create several users, assign them to groups, and then assign services to the groups.

In this section, you will learn how to create a user in the Users (USER) class in the POLICY domain of the Radia Database. You can follow the same steps to create a new Workgroups (WORKGRP) instance or Departments (DEPT) instance by substituting the appropriate class name.

In the following example, you will use the Radia System Explorer to create a new user (SSampson) in the USER class.

To create a new user

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

Note
The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

   The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click POLICY.

5. Right-click Users (USER).
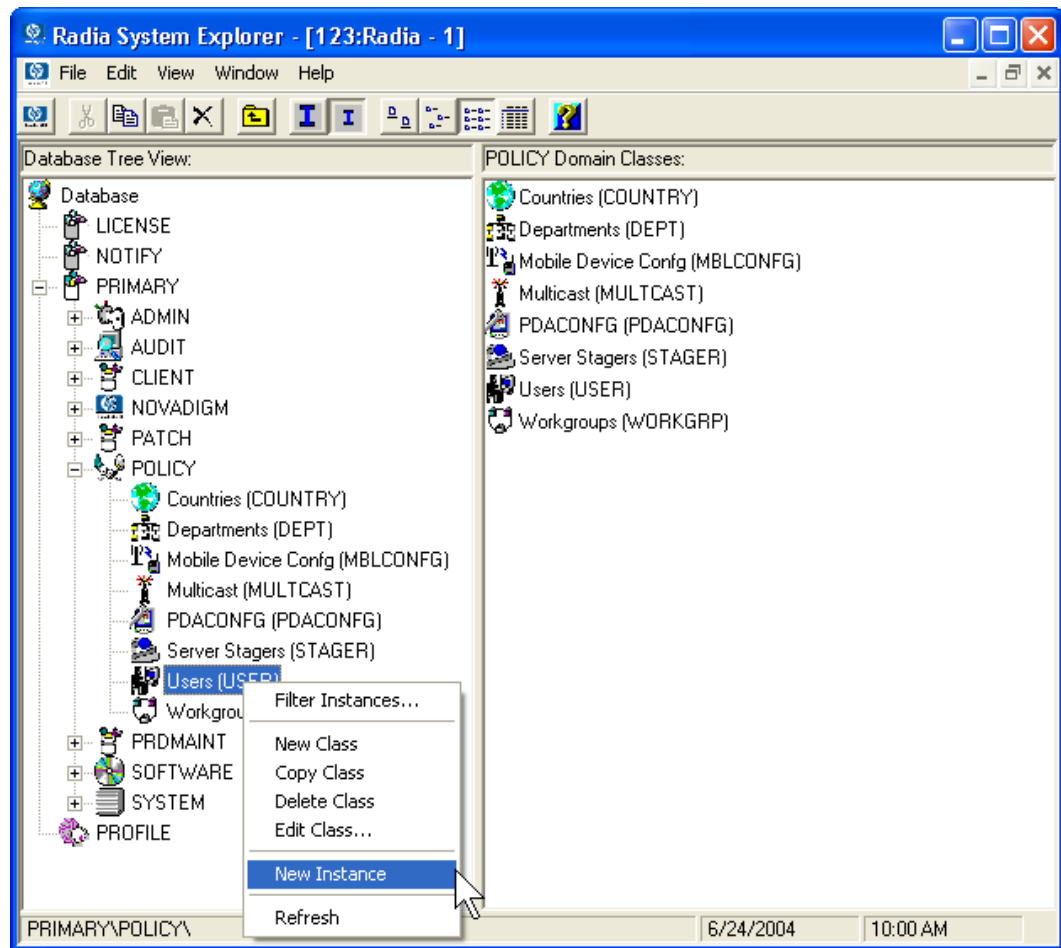
Figure 5.5 ~ Shortcut menu for the USER class.

6. Select New Instance.

7. In the **Create Instance** dialog box, type a display name (up to 25 characters) and an instance name (up to 25 characters).
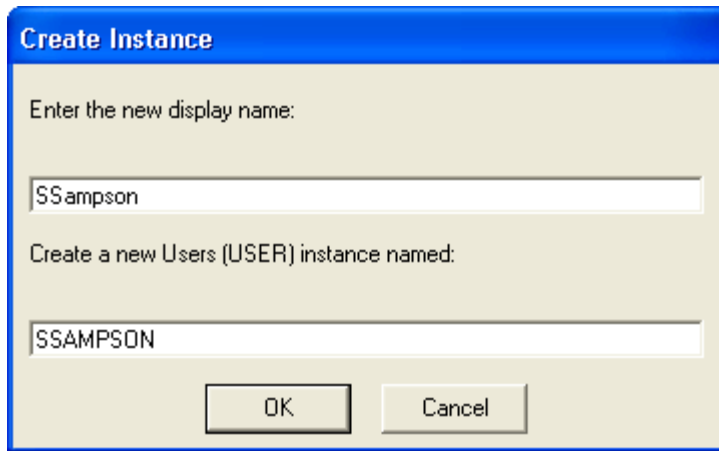
Figure 5.6 ~ Create Instance dialog box.

8. Click **OK**.

The user instance, SSampson, is created.
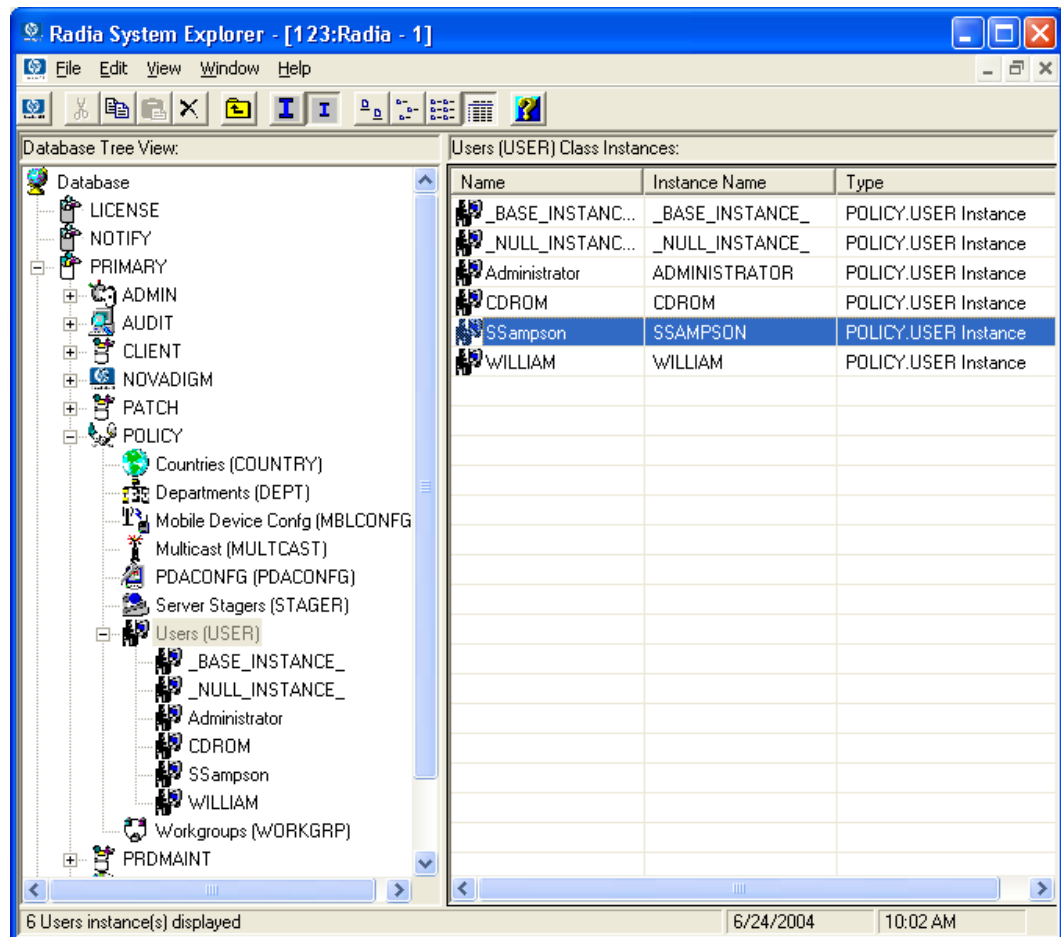
Figure 5.7 ~ The SSampson USER instance.

Assigning Users to Groups

If you have created several users, you might want to assign them to one or more groups. In the following example, we will use the Radia System Explorer to assign the user **SSampson** to the **Sales** department.

> **Note**
>
> The Sales instance, shown in the Departments (DEPT) class (see Figure 5.13 on page 252) may not appear in your Radia Database. To add this instance (or instances that are appropriate to your organization), follow the procedure *To create a new user* on page 241. However, instead of right-clicking USER, right-click the appropriate class, such as Departments (DEPT).

To assign a user to a group

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

   > **Note**
   >
   > The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

   The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **POLICY**.

5. Double-click **Users (USER)** to open the list of all user instances.

6. Right-click the user instance (in this example, SSampson) and select **Show Connections**.
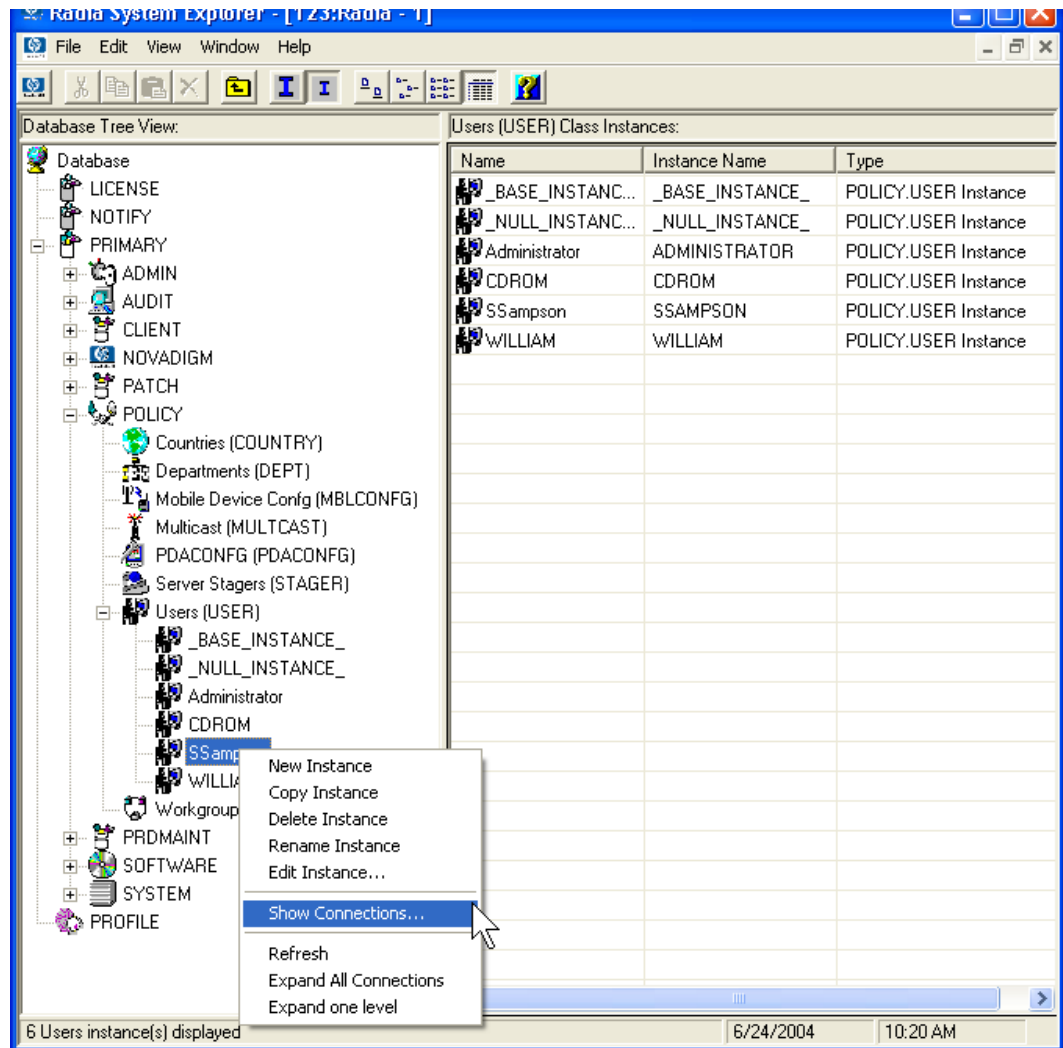
Figure 5.8 ~ Show the connectable classes for SSampson.

7. The **POLICY.USER Connections** dialog box opens. This dialog box displays a list of classes to which you can connect the selected instance.
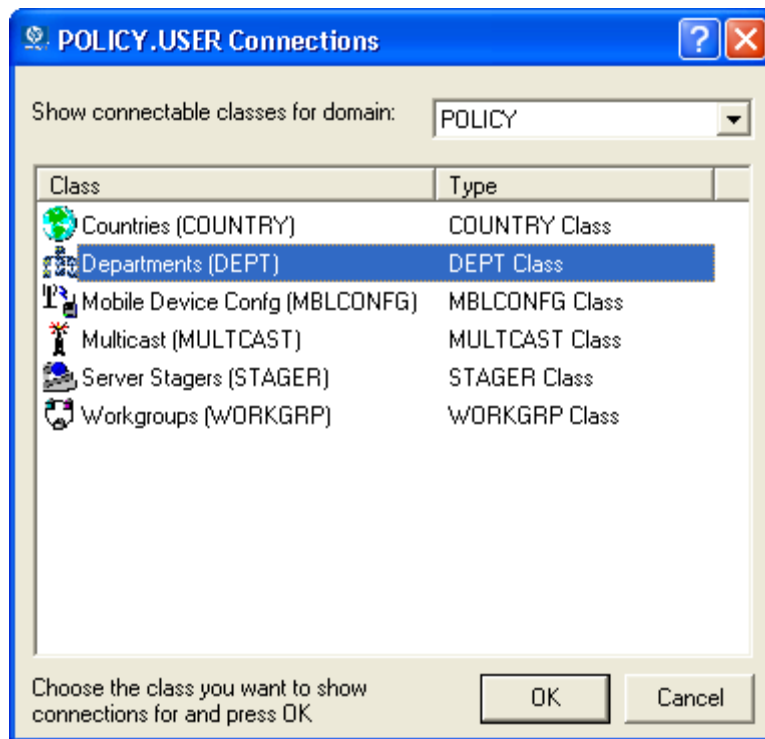
Figure 5.9 ~ Show Connectable Classes dialog box.

8. Select **Departments (DEPT)**, and then click **OK**.

The DEPT class instances appear in the list view. This allows you to easily make a connection between an instance in the DEPT class and an instance in the USER class.

9. Select the **Sales** instance from the list view, and then drag it to the appropriate USER instance (in this example, SSampson). When your cursor turns into a paper clip, release the mouse button.
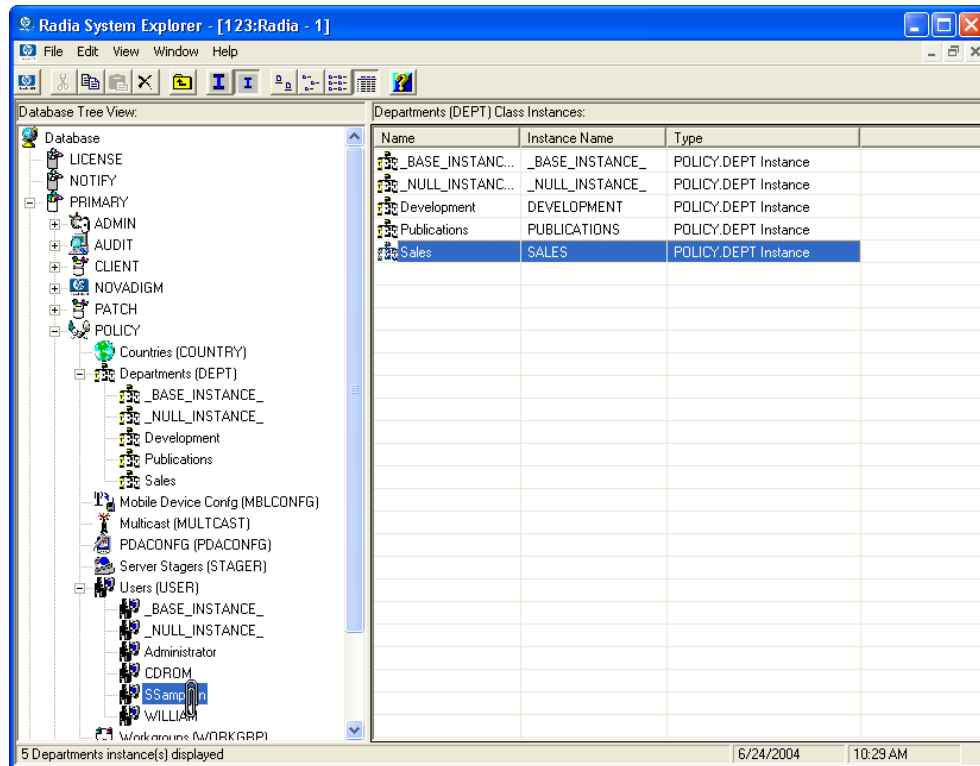
Figure 5.10 ~ Connect USERS.SSampson to DEPT.Sales.

10. The **Select Connection Attribute** dialog box opens.
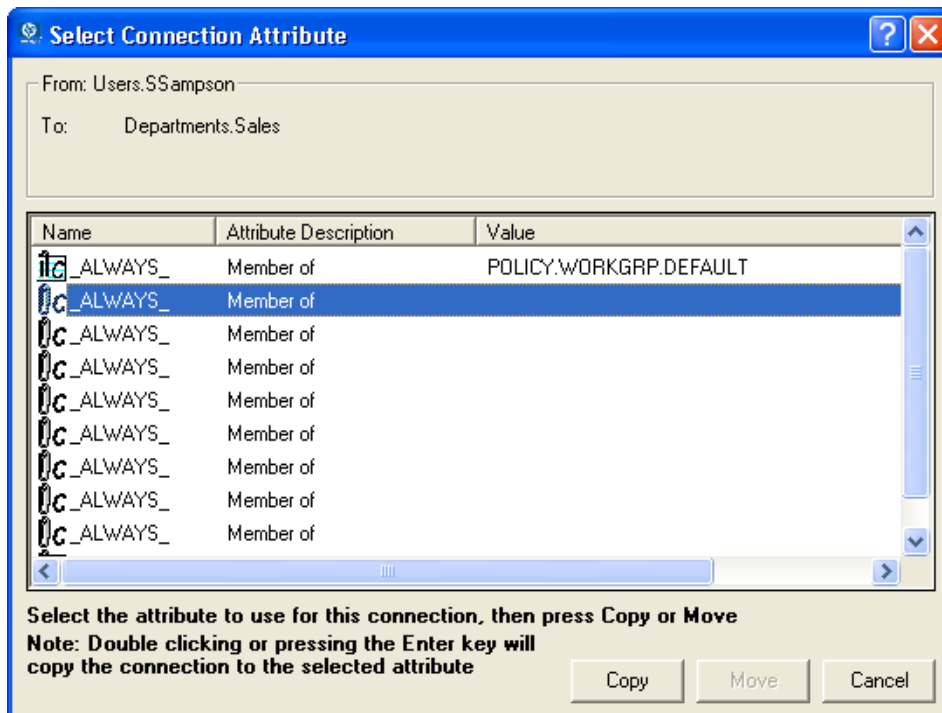
Figure 5.11 ~ Select the attribute for the connection.

11. Click **Copy** to create the connection from **Users.SSampson** to **Department.Sales**.

12. Click **Yes** to confirm the connection.

13. Click **OK** when you receive the confirmation that "SSampson has been connected to Sales."

Notice that **Sales** is now listed under the **SSampson** user instance, which indicates that SSampson is part of the Sales department.
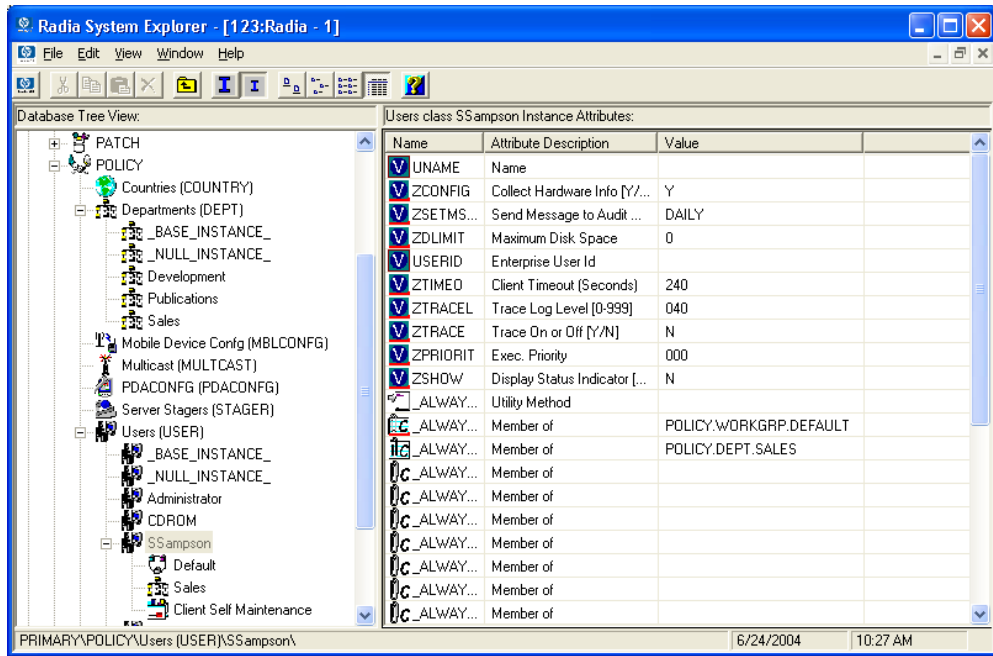
Figure 5.12 ~ SSampson is connected to the Sales department instance.

Connecting Services to Groups

Whether you are using an external policy source, or you are managing policy within Radia, you will need to define the services that your subscribers will receive.

> **Note**
>
> If you are using the Radia Policy Server, see the *Radia Policy Server* documentation on the HP OpenView web site for more information.

In this section, you will learn how to connect users and groups to the services that Radia will manage. In the following example, we will use the Radia System Explorer to authorize all subscribers in the Sales department for the WinZip application.

To connect the WinZip application to the Sales department

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

    The **Radia System Explorer Security Information** dialog box opens.

    > **Note**
    >
    > The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**. The **Radia System Explorer** window opens.
3. Double-click **PRIMARY**.
4. Double-click **POLICY**.
5. Double-click **Departments (DEPT)** to open the Departments class.
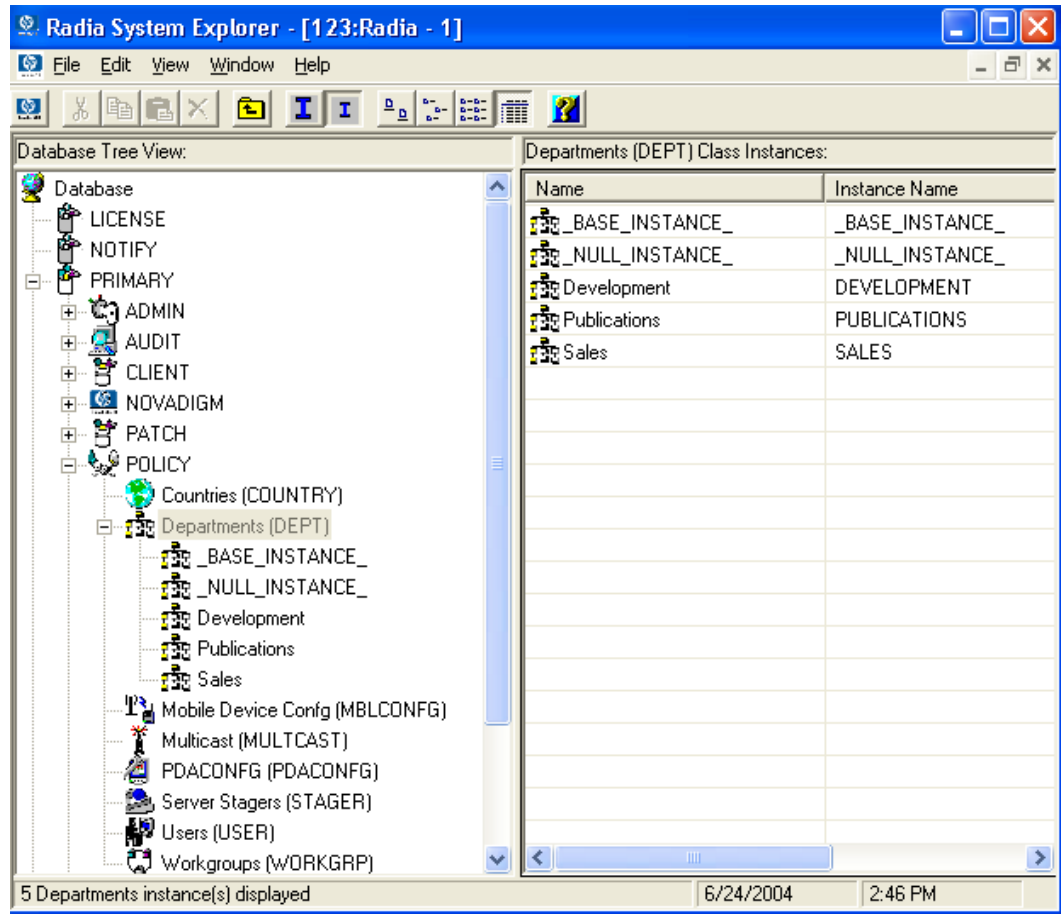
Figure 5.13 ~ The Departments (DEPT) class.

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 to which you can connect the selected instance.

7. From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.
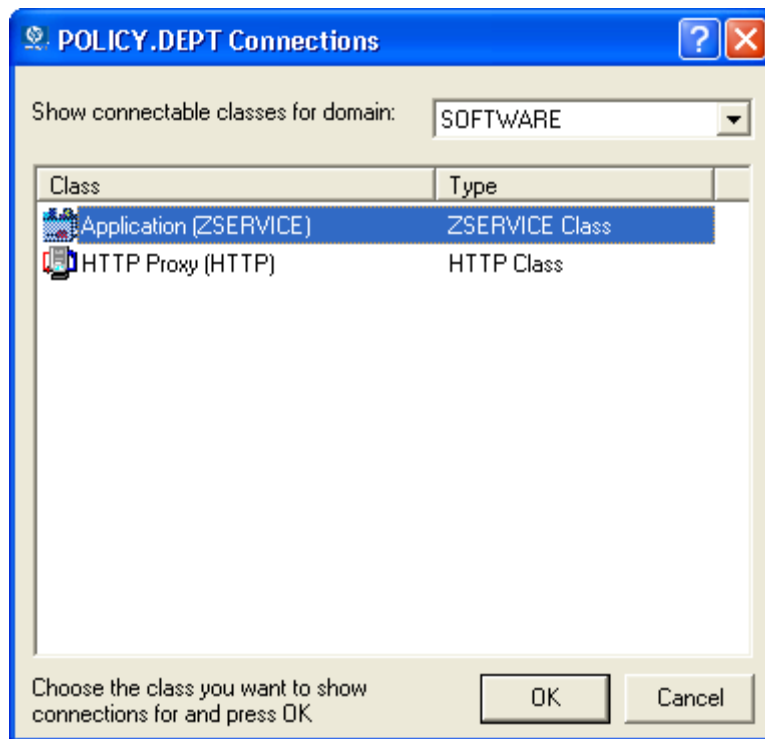
Figure 5.14 ~ The POLICY.DEPT Connections dialog box.

8. Click **Application (ZSERVICE)**, and then click **OK**. The instances in the ZSERVICE class appear in the list view.
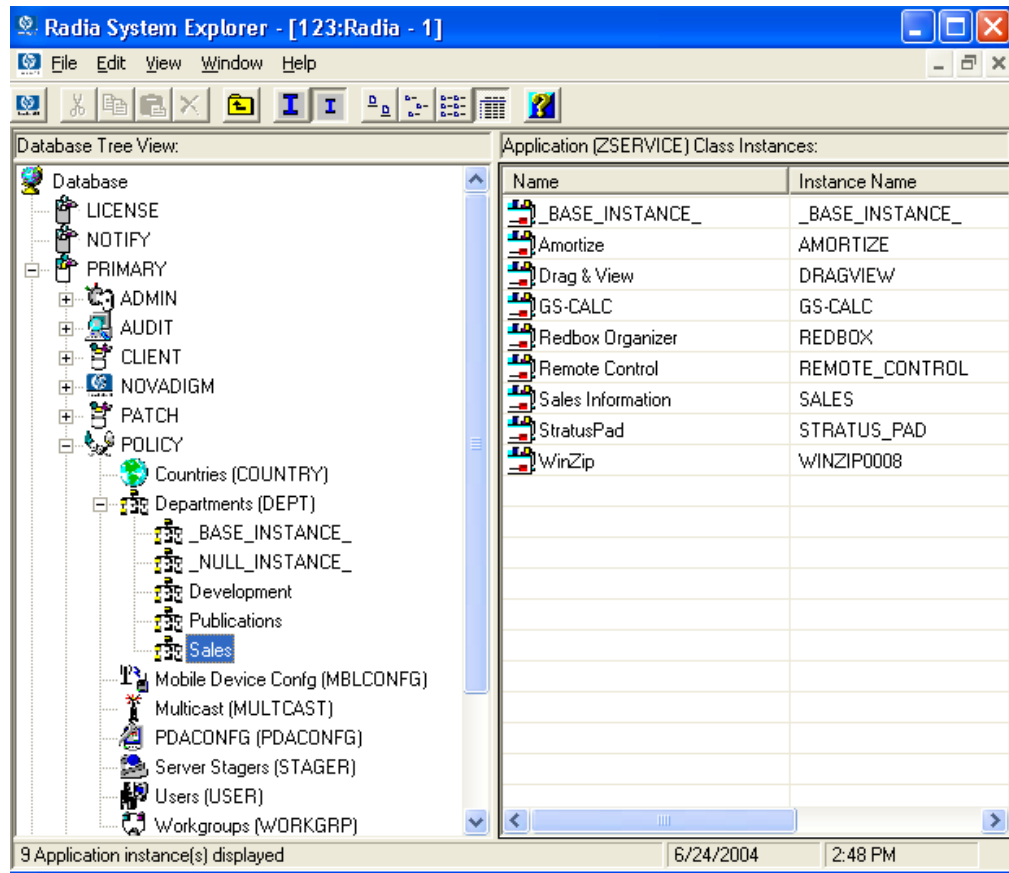
Figure 5.15 ~ ZSERVICE class instances in the list view.

9. Select the **WinZip** instance from the list view, and then drag it to the appropriate **Departments** instance (in this example, **Sales**). When your cursor turns into a paper clip, release the mouse button.
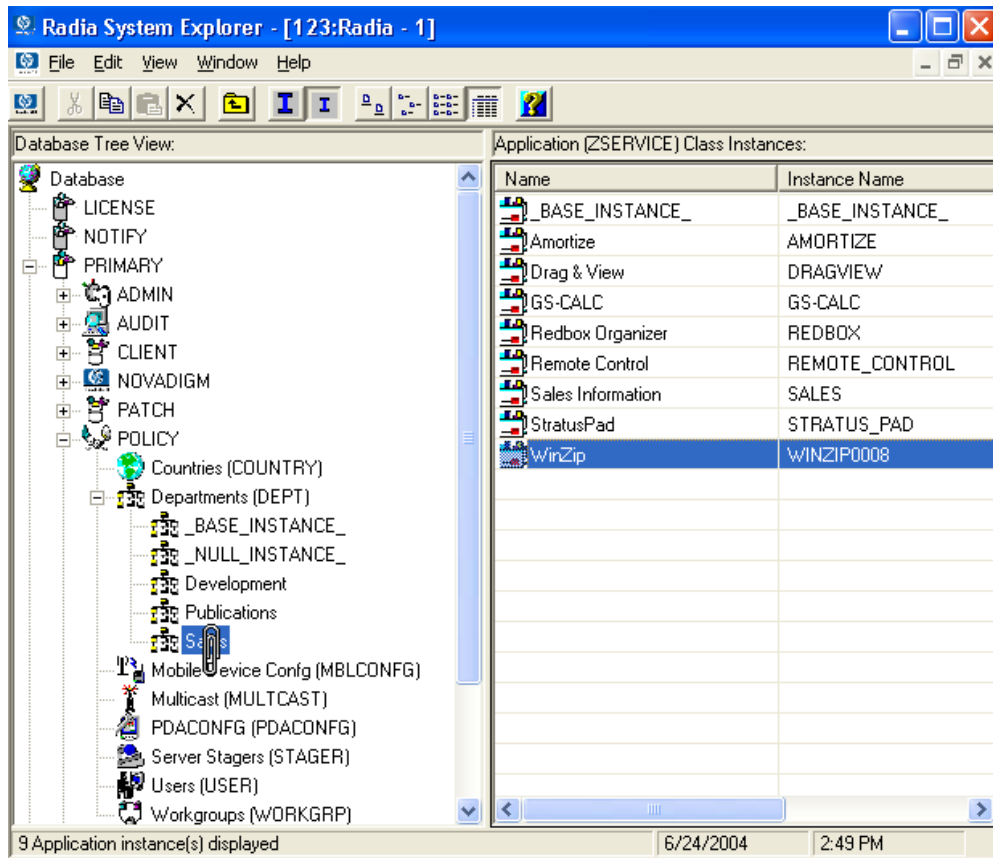
Figure 5.16 ~ Connect WinZip to Sales.

10. The **Select Connection Attribute** dialog box opens.
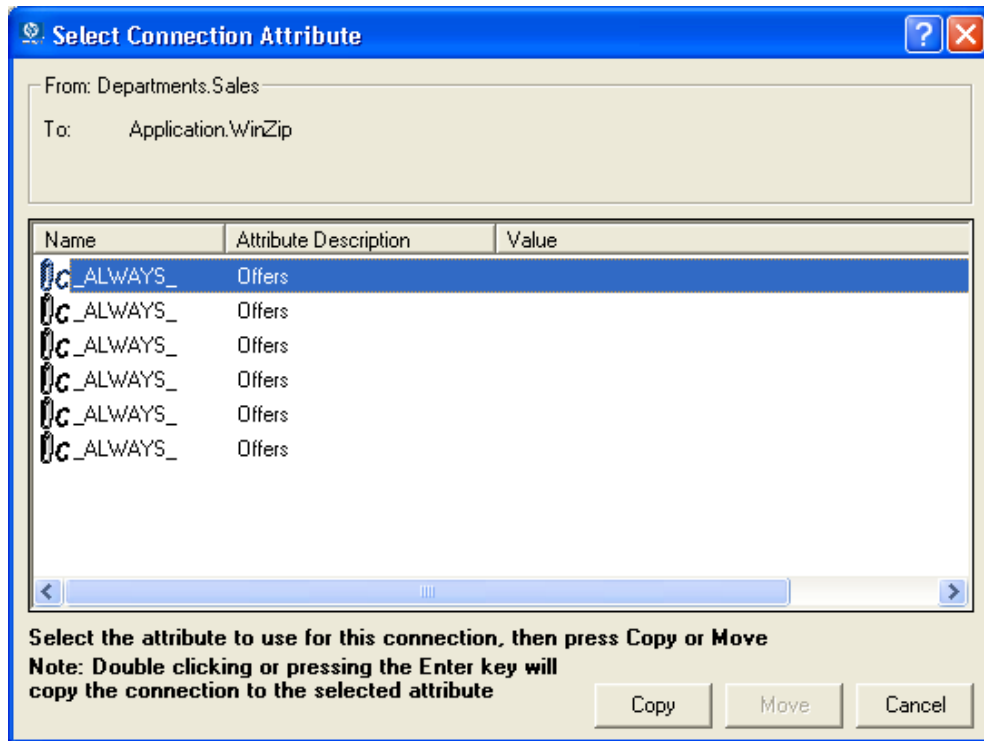
Figure 5.17 ~ Select the attribute for the connection.

11. Click **Copy** to create the connection from **Departments.Sales** to **Application.WinZip**.

12. Click **Yes** to confirm the connection.

13. Click **OK** when you receive the confirmation that "Sales has been connected to WinZip."

Notice that **WinZip** is listed under the **Sales** department instance, which indicates that the entire Sales department is now authorized to receive the WinZip application.
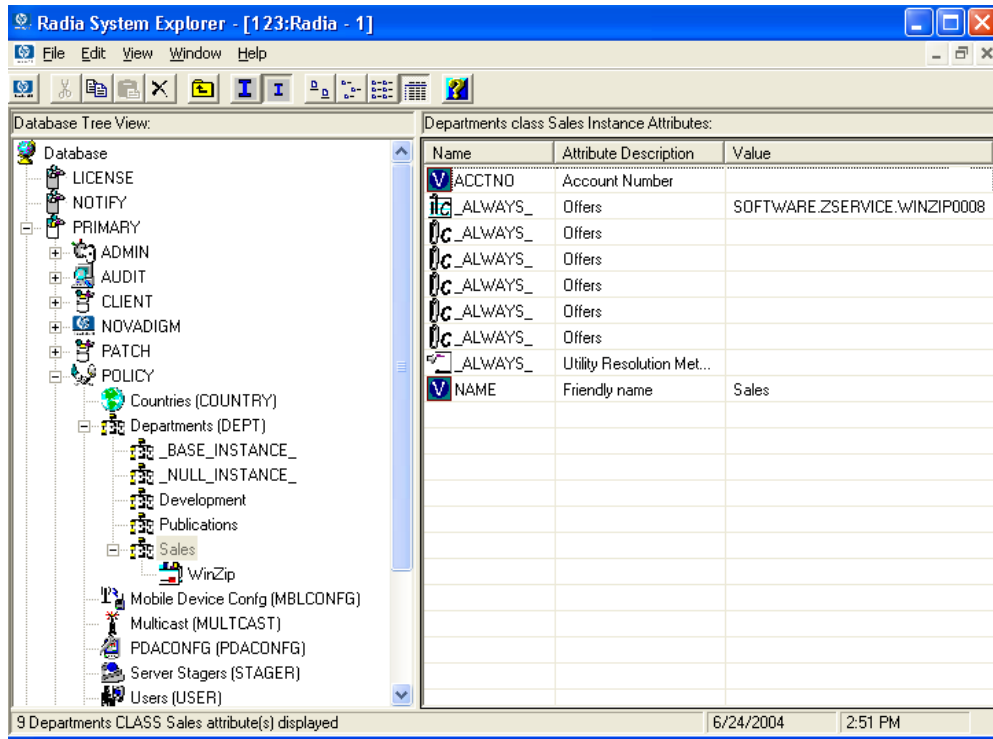
Figure 5.18 ~ The WinZip application is authorized for the Sales department.

In Figure 5.19 on page 258, notice that **SSampson**, listed in the **Users (USER)** class, is part of the **Sales** department. You can also see that the **WinZip** application has been authorized for the entire **Sales** department. Therefore, as long as SSampson is part of the Sales department, Radia will manage the WinZip application on his computer.
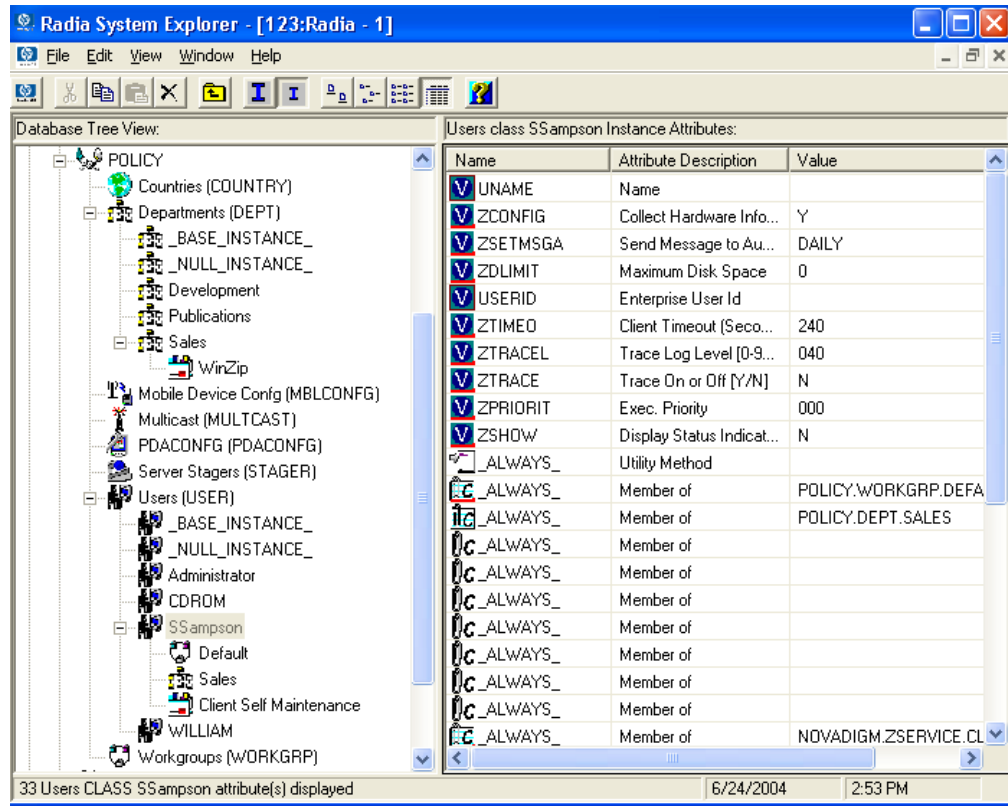
Figure 5.19 ~ Radia manages WinZip for the Sales department.

Whether you are using an external policy store, or managing policy within Radia, you can quickly modify the services that individuals are authorized for by manipulating the connections between services and groups, adding users to groups, or removing users from groups.

Summary

- Radia can integrate with your existing policy information.
- The Radia POLICY domain organizes subscribers into logical groups.
- You can create new users and assign them to groups.
- Assign the services to be managed by Radia to the appropriate groups.

Configuring Client Operations Profiles

At the end of this chapter, you will:

■ Understand the benefits of Radia Client Operations Profiles.

■ Know how to implement Radia Client Operations Profiles.

■ See a simple implementation example.

This guide covers the *standard* implementation for the Radia Software Manager. Although you will tailor this strategy to meet your organization's needs, it is recommended that you review this guide for a comprehensive understanding of the Radia Software Manager. This chapter introduces you to how to configure the operations of your Radia Client. It covers the CLIENT domain, and provides information on providing failover capabilities, designating servers for a client based on criteria you set, controlling trouble shooting settings, hardware scan settings and user interface settings.
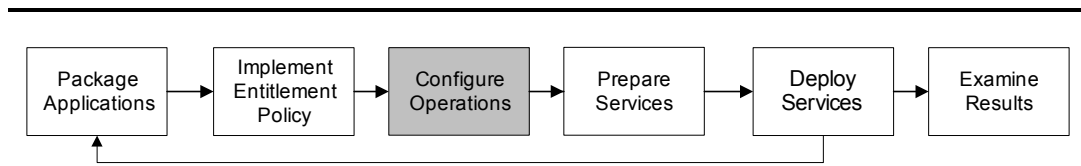
Figure 6.1 ~ Overview of the Radia Software Manager Guide.

Radia Client Operations Profiles

If you have multiple Radia Configuration Servers, Radia Proxy Servers, or want to store files for managing applications on a local CD-ROM, you may want to reconfigure the Radia Client before connecting to the Radia Configuration Server. Use Radia Client Operations Profiles to do this. Radia Client Operations Profiles, along with the Radia object oriented schema, allow you to control certain client behaviors based on any attribute from a client object. Benefits of using Radia Client Operation Profiles include, but are not limited to:

■   Failover capability for Radia servers.

■   Dynamic assignment and selection of a client computer's available servers based on network location, speed, or other criteria.

■   Defining which Radia Configuration Server to use based on its functional role, allowing for load balancing among Radia Configuration Servers.

■   Enhanced diagnostics capabilities.

> **Note**
>
> To use Client Operations Profiles, you must be using the Radia Application Manager version 3.1 or higher and the Radia Database version 3.1 or higher.

The CLIENT Domain

The CLIENT domain in the Radia Database controls Client Operations Profiles. It includes six classes with sample instances that you can use to configure your Radia Client computers operations. We provide an example of using the client computer's network location, stored in the ZCONFIG object to prioritize the downloadable locations for application data for each client computer. The six classes are:

■   **Core Settings (SETTINGS)**
Use an instance in the SETTINGS class to define how to use your Server Access Profile, to define scripts you want to use, and to set other global parameters.

■   **Diagnostics (DIAGS)**
Use instances in this class to override tracing levels set on the Radia Client.

■   **Hardware Scan Config (RADHWCFG)**
Use an instance in this class to control the type of hardware scan that the Radia Client should perform.

■   **Network Location (LOCATION)**
Use the LOCATION class to group users based on a location, such as their subnet.

■   **RSM UI Preferences (RADUICFG)**
Use instances in this class to manage the display of the Radia Software Manager User Interface.

- **Server Access Profile (SAP)**
  Use instances in the Server Access Profile (SAP) class to define Radia Configuration Servers and possible data access points for Radia managed services.

Recommendations

We make the following recommendations for using Client operations profiles.

- Use our Professional Services to help you implement this feature.
- Configure Client Operations Profiles *only* if you fully understand this process.
- Avoid single point of failure in all aspects, for servers of both types and for each role. Create redundancy where possible.
- Use base and null instances for unknown and new network addresses.

Implementing Client Operations Profiles

Use instances in the classes in the CLIENT domain to customize the profiles to meet your enterprise's needs. There are five major steps for implementation.

1. Identify Servers.
2. Create Server Access Profile instances.
3. Set criteria for Server Access Profile instances.
4. Set priorities for Server Access Profile instances.
5. Enable Client Operations Profiles.

Each of these steps is detailed in the sections below. Before beginning this procedure, a discussion of Server Types and Roles is required.

Understanding Server Types and Roles

A Server Access Profile (SAP) is a generic way to define all possible data access points for a service. A SAP can be a Radia Configuration Server, Radia Proxy Server, or CD-ROM drive. Radia Client Operations Profiles allow you to identify and prioritize data access points without the need to use additional customized scripts.

Before beginning this process, you must have an understanding of server types and roles. These are reflected, respectively, in the TYPE and ROLE attributes of the SAP class. A server can either be identified as an RCS or DATA type. Only a Radia Configuration Server can be identified as type RCS. A Radia Configuration Server, Radia Proxy Server, or CD-ROM drive can be identified as DATA in the TYPE attribute. Use DATA type only for servers from which the client will download applications.

In addition, each Radia Configuration Server can have a role, or function, specified in the ROLE attribute of the SAP class. Possible roles are:

■ **Client Operations Profiles (O)**
Use this Radia Configuration Server to get the client computer's Client Operations Profile.

■ **Service resolution (S)**
Use this Radia Configuration Server to resolve the client computer's services.

■ **Client self maintenance (M)**
Use this Radia Configuration Server to perform Client Self Maintenance.

■ **Reporting (R)**
Use this Radia Configuration Server for storing reporting objects from the client computer. These objects are stored in the PROFILE file in the Radia Database.

■ **Data download (D)**
Use this Radia Configuration Server to download application data to the client computer.

■ **All (A)**
Use this Radia Configuration Server for any of the roles listed above.

A Radia Proxy Server or CD-ROM can only serve the role of Data download (D). A Radia Configuration Server can serve any of the above roles.

Important Note

When a client is ready to download files, the client will first use servers with TYPE=DATA in order of priority set in the LOCATION class. If a server with TYPE=DATA, has a ROLE of anything other than D, it will be skipped and not used for data download.

If the client is still unable to download all the needed files after processing all servers with TYPE=DATA, then the client will begin processing servers of TYPE=RCS in order of priority. In order to use a Radia Configuration Server for data download, two conditions must be met:
- The SETTINGS.RCSDATA attribute for the client computer must be set to Y.
- The SAP.ROLE for the Radia Configuration Server must be either D or A.

Step 1: Identify Radia Servers

Identify your Radia Servers, and determine the type and role for each. Types are defined in the TYPE attribute of the SAP class. Roles are defined in the ROLE attribute of the SAP class. Your Radia Server can be a Radia Configuration Server, Radia Proxy Server, or CD-ROM. Radia Servers with TYPE=DATA, must have a role of Data download (D). Radia Configuration Servers, set to TYPE=RCS, can serve many roles. You will need to decide which roles your Radia Configuration Servers can perform.

Note

Only a Radia Configuration Server can be designated as anything other than data download. For all other SAP instances, if you specify anything other than data download, the SAP will be ignored.

Step 2: Create Server Access Profile Instances (SAP)

Use Radia System Explorer to create one SAP instance for each Server Access Profile. *Table 6.1 ~ Attributes of the SAP Class* on page 268 describes the attributes in the SAP class. We provide you with samples for each type of Server Access Profile in the Radia Database. *Figure 6.2 ~ Use Radia System Explorer to view SAP Samples* below displays the SAP samples we provide.
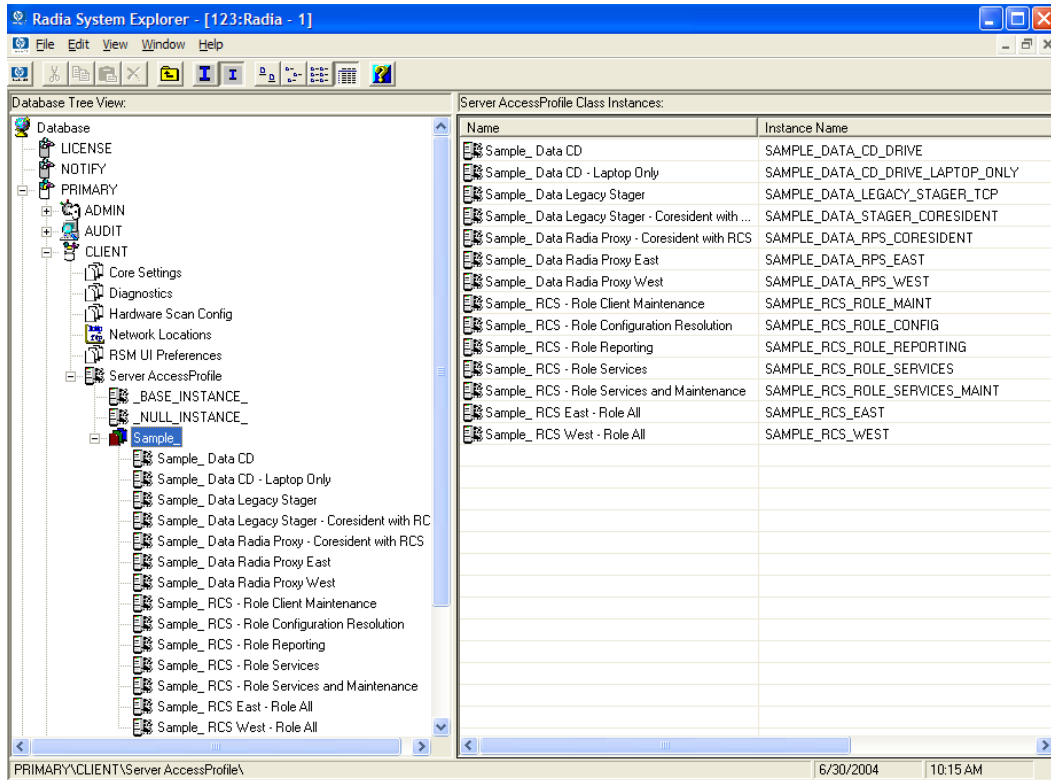
Figure 6.2 ~ Use Radia System Explorer to view SAP Samples.

Use Radia System Explorer to make a copy of the instance that most closely approximates the server type and role for which you need an SAP instance. After copying the instance, use *Table 6.1 ~ Attributes of the SAP Class* on page 268 to configure the instance for your enterprise.

> **Note**
>
> Instances in the SAP class are used before the IP specified in the **radskman** command line.

Table 6.1 ~ Attributes of the SAP Class

Attributes	Description
ZSTOP00n	**Expression Resolution Method** Use a ZSTOP expression to stop the process from completing if certain requirements are met. For example, you may want to prevent a laptop computer from using this SAP.
NAME	**Friendly Name** Friendly name of the SAP instance.
TYPE	**Type [RCS/DATA]** Specify the Type of Radia server. Set to **RCS** if using Radia Configuration Server. Set to **Data** for Radia Proxy Server or CD-ROM. If the client computer is unable to reach any of its Server Access Profile, then the client will default to the last known Radia Configuration Server.
URI	**Universal Resource Identifier** Create the Universal Resource Identifier to specify the Radia Configuration Server, or Radia Proxy Server. See *Table 6.2 ~ URI Examples* on page 270 for examples.
ROLE	**RCS Role A,O,S,M,R,D** Specifies the role of the SAP. Specify as many values as are needed separated by a comma. A blank or null value defaults to ALL. Possible values are A = all, O = Client Operations Profiles, S = Service Resolution, M = Client Self Maint, R = Reporting, D= Data Download. **Default**: The default value is A. **Note:** Only a Radia Configuration Server can be designated as anything other than type D. For servers where TYPE=DATA, if you specify anything other than D, that SAP instance will be skipped.
ENABLED	**Enable SAP [Y/N]** Specify if this SAP is enabled (**Y**) or disabled (**N**). If the variable is blank or non-existent, then this SAP is enabled. **Default**: The default value is **Y**.
TIMEOUT	**Communications Timeout (0-3200)s** Specify the timeout in seconds. This will override client timeout (ZMASTER.ZTIMEO) if it contains a valid numeric value. If blank, then use existing variable value on client.
PUSHBACK	**Push Back (0-999 retries)** Set to **0** to skip a Radia Configuration Server if the Radia Configuration Server pushes back on the Client Connect. Set to **1** to **999** for number of retries if the Radia Configuration Server pushes back. **Default**: The default setting is **0**.

Table 6.1 ~ Attributes of the SAP Class

Attributes	Description
THROTYPE	**Throttle [NONE/ADAPTIVE/RESERVED/]** Type of bandwidth throttling to use. Set to **ADAPTIVE** to yield to other services that are using the network. Set to **RESERVED** to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to **NONE** for no bandwidth throttling, and use the maximum available bandwidth. This attribute applies to Windows only. This will override client bandwidth throttling if it contains a valid value. If blank, then use existing variable value on the client computer.
BANDWDTH	**Bandwidth Percentage (1-99)** Specify the percentage of bandwidth to use between 1 and 99. If blank value or non-existent variable, then use all of the bandwidth. This attribute applies to Windows only. This will override client bandwidth setting if it contains a valid value. If blank, then use existing variable value on client computer.
STREAM	**Enable Streaming [Y/N]** Specify **Y** to use streaming. This will override the client setting in ZMASTER.ZNORSPNS. **Default**: The default value is N. **Caution**: Streaming is not suitable for all network environments. Consult your network administrator before setting this to Y.
PROXY	**Internet Proxy URI** *Do not modify.* The internet proxy URI through which the client will connect to the SAP. Maintained by client.
PRIORITY (&(LOCATION.SAPPRI))	**Selection Priority** *Do not modify.* The SAP obtains its priority by looking at the priority specified in the Location class.
PRODUCT	**Product Filter** Specify which types of Radia Clients can use this SAP instance. Specify multiple clients separated by a comma. Below are suggested identifiers for each Radia Client: Radia Application Manager: RAM Radia Inventory Manager: RIM Radia Software Manager: RSM Radia OS Manager: ROMS Radia Patch Manager: PATCH On your radskman command line, specify which products to filter by using the **product** parameter. For example, if this SAP should only be used by Radia Application Manager, then you may want to set this attribute to RAM. Then, set product to RAM on your radskman command line. **Default**: Blank means that all products can use this SAP instance.

Table 6.1 ~ Attributes of the SAP Class	
Attributes	**Description**
FILTER	**Filter Expression [Obj.Var = Value]**
	Use this attribute to filter the SAP based on any available object attribute. For example, if you only wanted to use this SAP for a specific service, specify APPINFO.ZOBJNAME=GS-CALC.
	Note: The ZSERVICE object is not available during installation. Use the APPINFO object instead. Appinfo, located in the service's LIB directory, is a copy of the service's instance from the ASERVICE object.

Note

If you want to override the use of the SAP object, add the RCSURI parameter to the **radskman** command line. RCSURI should be in the same format as the Universal Resource Identifier. For the syntax of this parameter, see *Table 6.2 ~ URI Examples* below.

Creating the Universal Resource Identifier

For each instance of the SAP class, you will need to identify a URI (Universal Resource Identifier) for the Server Access Profile (SAP). RFC 1630 proposes the following format:

<scheme>:<scheme specific format>

where the scheme is usually the network protocol such as HTTP or TCP. If the scheme-specific format has slashes, it indicates a hierarchical path. Universal Resource Locator (URL) is a form of URI where the scheme specific format is defined as:

//<user>:<password>@<host>:<port>/<url path>

Details on how to access the specified resource are defined in the URL path. The most prevalent form is: //<host>:<port>/<url path>. Examples for specifying the URI are shown in *Table 6.2 ~ URI Examples* below:

Table 6.2 ~ URI Examples	
SAP Type	**URI and TYPE attributes are set to:**
Radia Configuration Server over TCP/IP using default port of 3464	URI = tcp://novarcs:3464 TYPE = RCS
Radia Configuration Server over TCP/IP using port 7800	URI = tcp://novarcs:7800 TYPE = RCS
Radia Configuration Server using SSL on port 443	URI = tcps://novarcsssl:443 TYPE = RCS

Table 6.2 ~ URI Examples

SAP Type	URI and TYPE attributes are set to:
Radia Proxy Server using HTTP	URI = http://novarps:3466 TYPE = DATA
CD-ROM	URI = file://&(ZCONFIG.ZHDWCDDR)|/DATA/ TYPE = DATA

Step 3: Set Criteria for each SAP Instance

After creating your SAP instances, you must decide how you are going to segment your enterprise. You may want to assign an SAP to a client computer based on its subnet. If so, use Radia System Explorer to create one Location instance for each subnet. In the sample database, there are two locations, Sample_Location East and Sample_Location West.

The ZCONFIG object for a client computer includes an attribute called NETLOC. The ZCONFIG.NETLOC variable identifies the client computer's subnet using underscores instead of periods. You may want to name the Location instance based on possible subnets of your client computers so that you can easily connect a user to their appropriate LOCATION instance based on their subnet. For example, if you have a subnet of 10.10.10.1, create a LOCATION instance called 10_10_10_0. *Step 5: Enable Client Operations Profiles* on page 277 shows you how to connect the subscribers to the appropriate location class using the client computer's ZCONFIG.NETLOC variable.

Step 4: Set Priority for each SAP for each Location

Use instances in the LOCATION class to define the priorities of your Server Access Profile based on location criteria. *Figure 6.3 ~ View the sample LOCATION instances* below shows one of the samples that we provide. The priority for a Server Access Profile is defined directly above the connection to that SAP instance in the SAPPRI attribute. Lower numbers have a higher priority. For example, SAP.SAMPLE_RCS_EAST has a priority of 10, and SAP.SAMPLE_DATA_RPS_EAST has a priority of 40.

Note

The SAPPRI can be any whole number from 01 to 99. The SAPs do not need to be listed in the LOCATION instance in their priority order.
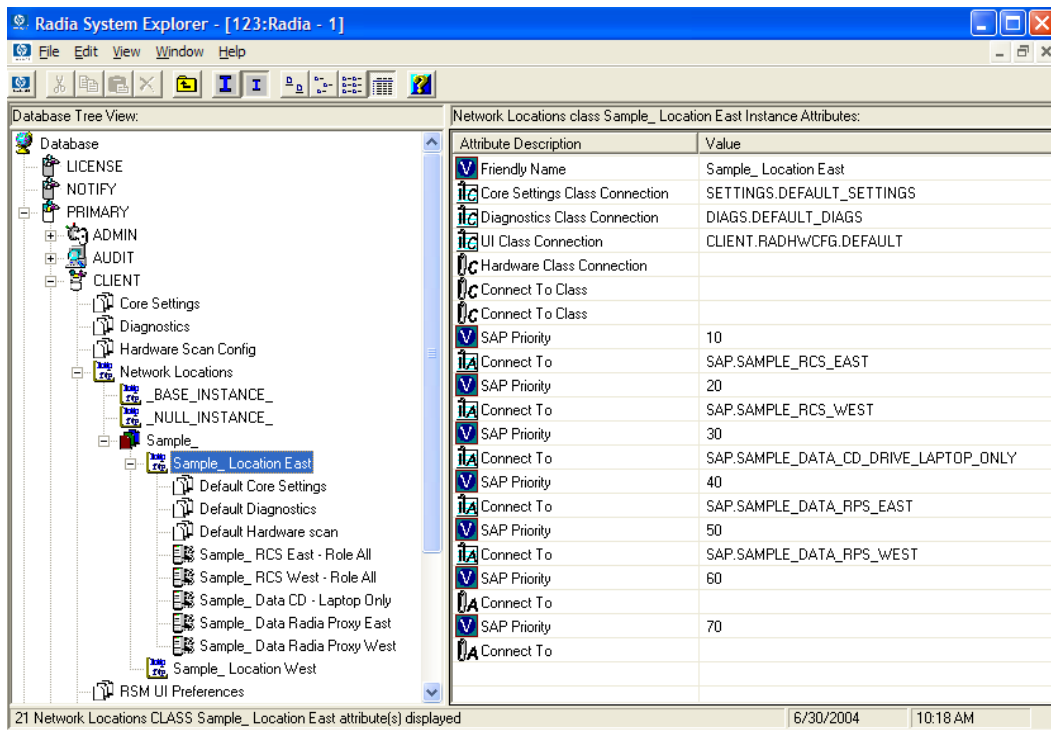
Figure 6.3 ~ View the sample LOCATION instances.

Table 6.3 ~ Attributes of the LOCATION Class describes the attributes of an instance in the LOCATION class. For example, if you want your SAMPLE_RCS_ WEST to be used before your

SAMPLE_RCS_ EAST, increase its priority to 7. To do this, change the SAPPRI for
SAMPLE_RCS_WEST to 7 as shown in *Figure 6.4 ~ Increase the priority of SAMPLE_RCS_WEST*
below.
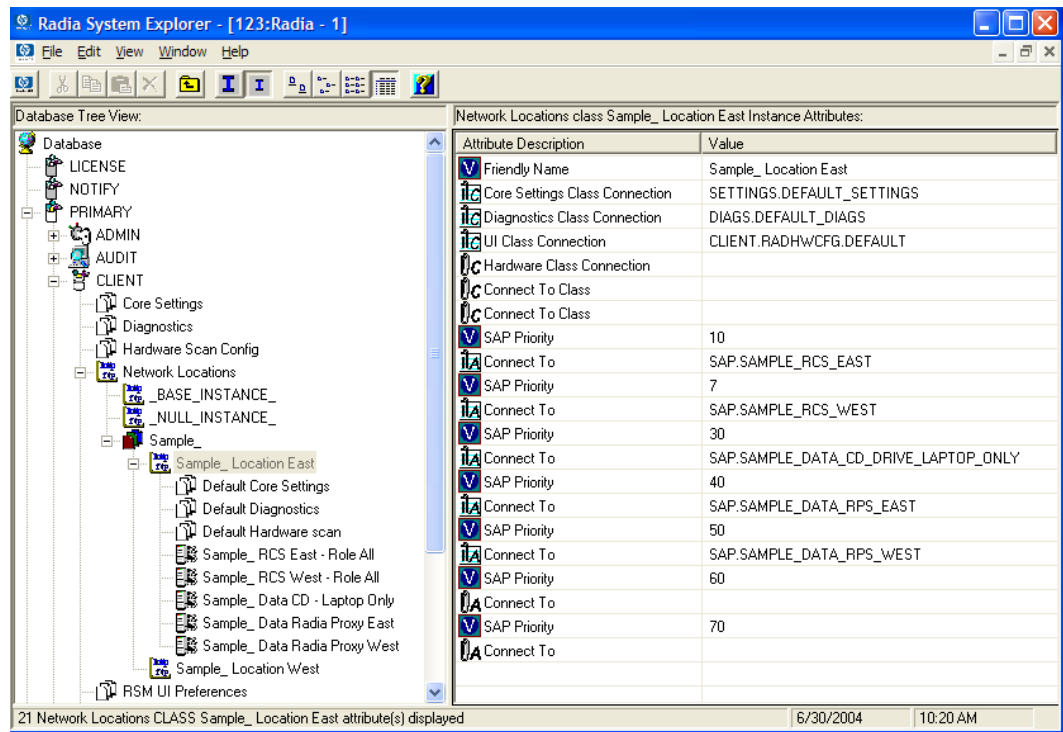
Figure 6.4 ~ Increase the priority of SAMPLE_RCS_WEST.

Table 6.3 ~ Attributes of the LOCATION Class	
Attribute	**Description**
COPLNAME	**Friendly Name** Type the friendly name of the instance.
ALWAYS	**Core Settings Class Connection** Specify an instance in the SETTINGS class. **Default**: The default connection is SETTINGS.DEFAULT_SETTINGS.
ALWAYS	**Diagnostics Class Connection** Specify an instance in the DIAGS class. **Default**: The default connection is DIAGS.DEFAULT_DIAGS.

Table 6.3 ~ Attributes of the LOCATION Class

Attribute	Description
ALWAYS	**UI Class Connection**
	Specify an instance in the RADUICFG class.
ALWAYS	**Hardware Class Connection**
	Specify an instance in the RADHWCFG class.
ALWAYS	**Class Connection**
	Specify an instance in any class to connect to this Location instance.
SAPPRI	**SAP Priority**
	Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.
	Default: The SAP referenced in the connection below this instance has a priority of 10 by default.
ALWAYS	**Connect to**
	Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 10.
SAPPRI	**SAP Priority**
	Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.
	Default: The SAP referenced in the connection below this instance has a priority of 20 by default.
ALWAYS	**Connect to**
	Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 20.
SAPPRI	**SAP Priority**
	Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.
	Default: The SAP referenced in the connection below this instance has a priority of 30 by default.
ALWAYS	**Connect to**
	Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 30.
SAPPRI	**SAP Priority**
	Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute.
	Default: The SAP referenced in the connection below this instance has a priority of 40 by default.
ALWAYS	**Connect to**
	Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 40.

Table 6.3 ~ Attributes of the LOCATION Class

Attribute	Description
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. **Default**: The SAP referenced in the connection below this instance has a priority of 50 by default.
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 50.
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. **Default**: The SAP referenced in the connection below this instance has a priority of 60 by default.
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 60.
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. **Default**: The SAP referenced in the connection below this instance has a priority of 70 by default.
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 70.

Step 5: Enable Client Operations Profiles

There are two phases to this step. First, you will need to create a process on the Radia Configuration Server so that the objects associated with Client Operations Profiles are resolved. Second, you will need to enable the client computer to use Client Operations Profiles.

Enable on the Radia Configuration Server

To enable Client Operations Profiles, you must create a new instance called RADSETUP in the Radia Processes (PROCESS) class in the SYSTEM domain. After creating the instance, make a class connection to CLIENT.LOCATION.&(ZCONFIG.ZNETLOC). This will map the user to its appropriate Location instance based on the ZNETLOC attribute in the client computer's ZCONFIG object. The ZNETLOC attribute identifies the client computer's subnet using underscores instead of periods to separate the four octets. Instances of the LOCATION class must be defined to match each of the subnets in your environment to be able to dynamically assign Client Operations Profiles configuration settings based on locations within your network. *Figure 6.5 ~ Create a SYSTEM.PROCESS.RADSETUP Instance* on page 278 shows an example.
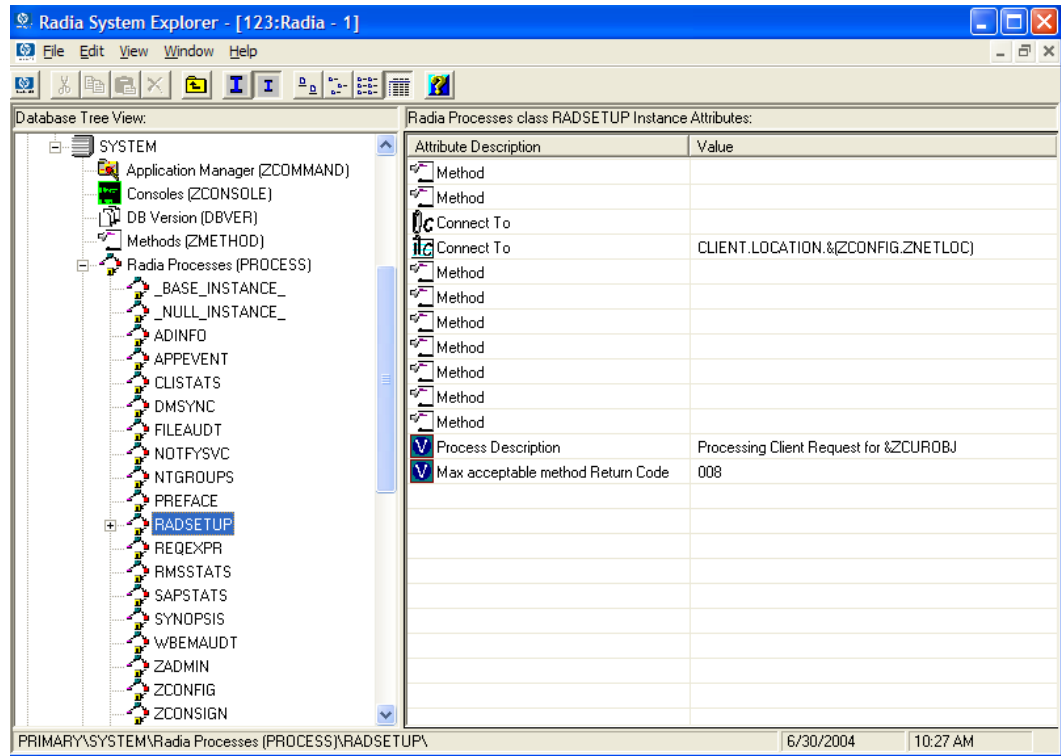
Figure 6.5 ~ Create a SYSTEM.PROCESS.RADSETUP Instance.

Enable on the Client

By default, Client Operations Profiles are disabled on client computers for backwards compatibility with older version of Radia. There are three ways to enable Client Operations Profiles on the client computer. Choose your method based on whether the Radia Client has already been installed, and the method that suits your needs best.

If you have not already installed the Radia Client,

■ You can customize the install.ini to add the COP variable to the RADSETUP object. To do this, add one line to the [objects] section in the install.ini. The figure below shows an example of the section with a new line. See *[OBJECTS] Section of INSTALL.INI* on page 45 for more details.

```
[Objects]
; Set Radia object attribute values
; A value of _NONE_ will set the attribute to blank
;
RADSETUP_COP=Y
;ZMASTER_ZDSTSOCK=
;ZMASTER_ZIPADDR=
;ZMASTER_ZNTFPORT=3465
```

Figure 6.6 ~ Add RADSETUP line to Objects section of Install.ini.

If you want to enable Client Operations Profiles on already existing Radia Clients:

■   You can use a rexx method, **initmeth.rex** to add and set the COP variable in the RADSETUP
    object. **Initmeth.rex** runs each time a "First Refresh Catalog" is called and can build the
    RADSETUP object with COP=Y to enable Client Operations Profiles. The figure below shows
    the lines to add to your **initmeth.rex**. Be sure to deploy the updated **initmeth.rex**.

```
/* Sample INITMETH.REX to Enable the COP */
call edmget('RADSETUP',0)
RADSETUP.COP = 'Y'
call edmset 'RADSETUP'
```

Figure 6.7 ~ Add these lines to initmeth.rex.

■   You can use the COP parameter of **radskman** to enable or disable Client Operations Profiles.
    To enable, add COP=Y to your list of parameters for **radskman**. *This will only enable or
    disable Client Operations Profiles for this Client Connect.* Use **initmeth.rex** as shown above
    to enable COP for all Client Connects or create a variable in CLIENT.SETTINGS called COP,
    and set the value to Y. Ultimately, both of these methods will create a COP attribute in the
    RADSETUP object with a value of Y. Alternatively, if you need to disable Client Operations
    Profiles after enabling it, run **radskman** with COP=N to disable for *that* Client Connect only.

Additional Classes in the CLIENT Domain

There are two additional classes in the CLIENT domain used for customizations and diagnosis. Define your own scripts to be used during the Client Connect with the Core Settings (SETTINGS) class. Set tracing levels and use other diagnostic tools by configuring the attributes in the Diagnostics (DIAGS) class

Core Settings (SETTINGS)

Use an instance in the SETTINGS class to define how to use your Server Access Profile, define scripts you want to use in pre-configuration processing, and set other global parameters.

Note

If similar attributes exist in both the SETTINGS and SAP classes, the attribute in the SAP class will be used.

Table 6.4 ~ Attributes of the SETTINGS Class

Attributes	Description
COPSNAME	**Friendly Name** Type the friendly name of the instance.
SAPPING	**Ping all SAP [Y/N]** Set to **Y** if the client should ping all of the SAPs. If EQUISORT is set to S, then you must set SAPPING to Y. A result reflecting the speed of the connection will be returned and stored in the SPEED attribute in the SAPSTATS object. **Default**: The default setting is N.
PUSHBACK	**Push Back (0-999 retries)** Set to 0 to skip a Radia Configuration Server if the Radia Configuration Server pushes back on the Client Connect. Set to 1 to 999 for number of retries if the Radia Configuration Server pushes back. **Default**: The default setting is 0.
EQUISORT	**Secondary SAP Priority [R/S]** If several SAP instances have the same priority, set this to **R** to randomly select which one to use. Set to **S** to use the SAP with faster network speed. SAPPING must be set to **Y** to use EQUISORT= S. Use **R** for workload balancing. **Default**: The default setting is **R**.

Table 6.4 ~ Attributes of the SETTINGS Class

Attributes	Description
USELSAP	**Use Last SAP [N/Y]** Set this Y to specify that the last SAP used in this Client Connect should be the SAP used for all remaining services to be resolved. Use of SAPs with type of DATA is at the service level. If set to N, then the client will go through the SAPs in priority for each service. During a Client Connect, if a service has to go to a lower priority SAP to complete the data download, decide if you want the remaining services to continue from this SAP (USELSAP=Y) or go back to the highest priority DATA SAP to search for files for the next service (USELSAP=N) and to continue through the SAP priorities. **Default**: The default setting is Y.
RCSDATA	**Download DATA from RCS [Y/N]** After using all of the TYPE = DATA SAPS, if all the needed data has not been downloaded then specify Y to go to SAPs with TYPE = RCS. If you do not want the client computers to use Radia Configuration Servers, set RCSDATA to N. **Default**: The default setting is Y.
ADINFO	**Query Active Directory Info [Y/N]** Specify Y if you want to collect the client computer's active directory information. The information is stored in the ADINFO object in the RADSETUP directory. The default location for the RADSETUP directory is <System Drive>:\Program Files\Novadigm\Lib\RADSETUP. This information will be sent to the Radia Configuration Server for all resolution processes. **Default**: The default setting is Y.
ZGRPINFO	**Query NT User Group Info [Y/N]** Specify Y if you want to collect the client computer's Windows NT user group information. This information will be reflected in the NTGROUPS object in the RADSETUP directory. The default location for the RADSETUP directory is <System Drive>:\Program Files\Novadigm\Lib\RADSETUP. This information will be sent to the Radia Configuration Server for all resolution processes. **Default**: The default setting is Y.
LSCRIPT	**Disable Connect on UI Reboot [Y/N]** If you have set a service to perform an immediate reboot and you run **radskman** from a login script, set this to **Y** to run **radskman** from the login script. If you have set a service to perform an immediate reboot, and you want **radskman** to be restarted in the User context when a user logs, set this to **N**. In other words, if your users are configured to connect to Radia Configuration Server when they log in, set this to N. If you want to do an immediate reboot with context=u, and you want the user to re-establish connection with the Radia Configuration Server, then set LSCRIPT=Y. For more information on reboot options, see the section Restarting the Client Computer. **Default**: The default setting is Y.

Table 6.4 ~ Attributes of the SETTINGS Class

Attributes	Description
ALWAYSD	**Always Download CFG Objects [Y/N]**
	Set to **Y** to always download pre-configuration objects. This guarantees that your SAP or persistent objects are downloaded even if nothing has changed. If your SAP client object is corrupted for any reason, then it will be re-downloaded even if the desired state didn't change. In addition, if one of the variables is a substitution then you will download the object with the new values since a variable change by substitution doesn't change the desired state.
	Default: The default setting is Y.
ALWAYSS	**Always Upload CFG Objects [Y/N]**
	Set to **Y** to always upload all objects in the RADSETUP directory to the Radia Configuration Server. The default location for the RADSETUP directory is <System Drive>:\Program Files\Novadigm\Lib\RADSETUP. Set to **N** to prevent the objects from being sent.
	Default: The default setting is Y.
EXBSETUP	**Pre Config Resolution Script**
	Specify a script to run before pre-configuration processing. This script must be in the client computer's IDMSYS directory. The default location is <System Drive>:\Program Files\Novadigm.
	Default: The default setting is PRESETUP.REX.
EXASETUP	**Post Config Resolution Script**
	Specify a script to run after pre-configuration processing. This script must be in the client computer's IDMSYS directory. The default location is <System Drive>:\Program Files\Novadigm.
CMETHOD	**Post Catalog Script**
	Specify a script that can run after catalog resolution, but before service processing.
EXBOUTBX	**Pre Outbox Script**
	Specify a script that can run after service processing, but before the objects in the outbox are flushed to the Radia Configuration Server.
EXBEXIT	**Post Connection Script**
	Specify a script to execute before **radskman** ends. If you are doing a customized reboot process, this is where you would specify it. This script must be in the client computer's IDMSYS directory. The default location is <System Drive>:\Program Files\Novadigm.
	Note: Client Operations Profiles must be enabled on the client for the EXBEXIT to be used. If Client Operations Profiles are not enabled, the EXBEXIT will be ignored.
TIMEOUT	**Communications Timeout (0-3200)s**
	Specify the timeout in seconds for the Server Access Profile (SAP). This will override the client timeout (ZMASTER.ZTIMEO) if it contains a valid numeric value. If the value is blank, then the client will use the existing timeout value on client.
THROTYPE	**Throttle [RESERVED/ADAPTIVE/NONE/]**
	Type of bandwidth throttling to use. Set to **ADAPTIVE** to yield to other services that are using the network. Set to **RESERVED** to allow for a specific reservation of the bandwidth. It is the maximum percentage of network bandwidth to use. Set to **NONE** for no bandwidth throttling, and use the maximum available bandwidth. **NONE** is the default. This attribute applies to Windows only.

Table 6.4 ~ Attributes of the SETTINGS Class

Attributes	Description
BANDWDTH	**Bandwidth Percentage (1-99)**
	Specify the percentage of bandwidth to use between 1 and 99. If the value is blank or the variable does not exist, then all of the bandwidth will be used. This attribute applies to Windows only.
RADTRAY	**Radtray Command Line Arguments**
	Set command line arguments you want to use for the Radia System Tray. Specify **Y** for the first argument to enable the Radia System Tray, and **N** to disable it. If set to **Y**, you can then specify other parameters separated from the **Y** with a comma. Possible parameters are:
	/C — Show the Radia System Tray in console mode when it starts.
	/NOCANCEL — Hide the Cancel button.
	/NOPAUSE — Hide the Pause button.
	/D — Add debug message to the log for troubleshooting.
	Example: **Y, /C /NOPAUSE** enables the Radia System Tray in console mode and does not display the PAUSE button.
USEDEFS	**Use Default SAP [Y/N]**
	If a SAP cannot be found for the needed ROLE, specify Y to default to the Radia Configuration Server set on the command line.
DEFROLE	**Default SAP ROLE (A,O,S,M,R)**
	Specify roles for the Radia Configuration Server specified on the command line. If not specified, the ROLE is set to A (All), and the Radia Configuration Server will be able to perform any ROLE.
	Note: USEDEFS must be set to Y to use DEFROLE.
RAD2XUI	**Enable RADUI 2.x**
	Specify **Y** to view the vintage Radia UI dialogs. Use this if you are not using Radia System Tray or if you want a message to pop up on the screen in addition to the Radia System Tray.
RSTROPT	**Bandwidth Checkpoint Restart**
	Use this attribute to determine when a file is eligible for checkpoint restart based on calculated network bandwidth. This will apply to *all* files to be downloaded in this Client Connect. Specify eligibility in the format (Below Threshold limit, Network Threshold Value, Above Threshold). Suppose you set RSTROPT to 100KB, 86KB, 10MB. First the client calculates the network bandwidth. One of two scenarios will apply:
	• If the network bandwidth is under 86KB, the file size is compared to 100KB. If the file size is over 100KB, checkpoint restart is enabled for that file.
	• If the network bandwidth is over 86KB, the file size is compared to 10MB. If the file size is over 10MB, checkpoint restart is enabled for that file.
DISKFREE	**Minimum Free Disk Space Threshold**
	Specify a minimum of free disk space for Radia to maintain. If a service is over the limit, it will not be installed.

Table 6.4 ~ Attributes of the SETTINGS Class

Attributes	Description
REMUNINS	**Allow Remote Notify Uninstall [Y/N]** Specify **Y** to stop notifies from remote machines from uninstalling a service. This does not stop applications from being un-installed as part of a policy change if a normal Client Connect is started from a remote notify. The remove notify string must contain the text req="Un-install."
DETPROXY	**Internet Proxy detection [Y/N]** Set to **N** to skip running Internet proxy detection at the beginning of the Client Connect.
ACTMAINT	**Maintenance Activation [I/D/P]** The Radia maintenance module, upgrdmaint, processes all maintenance activities. Upgrdmaint can be launched by radskman immediately after the maintenance is staged or on an independent schedule. **Note:** The mnt parameter of radskman must be set to Y for maintenance to be processed. Set ACTMAINT to **I** (Immediate) to download maintenance files and immediately activate them. Radia Software Manager subscribers will receive a dialog box showing just an OK button that RSM needs to be updated. Radia Software Manager will close, install maintenance, and then restart. Set to **D** to defer maintenance activation. Maintenance files are downloaded, but not activated. To activate maintenance, you can call radsksman req="Self Maintenance" or call upgrdmaint directly using a timer or other method. Set to **P** to prompt Radia Software Manager users. A dialog box will display stating that maintenance is available, but the subscriber will be given the option to cancel. The files are downloaded, but not activated. The subscriber will be prompted again at the next check for maintenance by the Radia Software Manager Interface. P is the same as I for Radia Application Manager users. **Default**: I
SENDRPT	**Send Reporting Object [I/D]** Set to D to defer sending all reporting objects to Radia Configuration Server at the end of Client Connect. Usually, the reporting objects for each service, such as APPEVENT, CLISTATS, and ZSVCSTAT, are sent to the Radia Configuration Server immediately (I) after they are created. This requires multiple disconnects and reconnects to the Radia Configuration Server. **Default**: I

Diagnostics (DIAGS)

Use this class to override default trace settings on the Radia Client computer. In addition, you can set parameters for running the **radstate** program. **Radstate** is a diagnostic module designed to give an overview of the current state of the Radia Client. The information in the **radstate** output is based on data retrieved from numerous Radia Client objects. For additional information on **radstate**, see the HP OpenView web site, and search for "Client Diagnostic Utility."

> **Note**
>
> Instances of this class allow you to easily set tracing levels and set parameters for **radstate** for a particular user, machine, or group of users for troubleshooting purposes. These attributes were intentionally put into their own transient class for this purpose.
>
> To do this, set the _ALWAYS_ Diagnostics Class Connection in the _BASE_INSTANCE_ of the LOCATION class to DIAGS.&(ZCONFIG.ZHDWCOMP). Then, create an instance in the DIAGS class with the computer name of the Radia Client computer that you want to set the tracing for. If the machine name does not exist in the DIAGS class, then the settings in the DEFAULT_DIAGS instance will be used.

Table 6.5 ~ Attributes of the DIAGS Class

Attribute	Description
COPDNAME	**Friendly Name** Type the friendly name of the instance.
RADSTATE	**Command String for radstate** This will run **radstate** with the parameters specified in this attribute. The base instance of the DIAGS class is set to VO, which will run **radstate** in verbose mode, building the ZRSTATE and ZRSTATES objects. See the documentation on **radstate** for more information. If no parameters are specified, **radstate** will not run. **Radstate** must exist in the IDMSYS directory. You only need to specify the parameters for **radstate**, not the **radstate** executable. Refer to the document on **radstate** for additional information.
ZTRACE	**Communication Tracing [Y/S/N]** Specify **N** to turns off communication buffer tracing. Tracing is off by default. Specify **S** to provide summary communication buffer information to the client log. This includes number of records read and written and the type of records processed. Specify **Y** to provide full communication buffer information to the client log. All data transmitted and received will be echoed to the client log file **Caution:** Setting ZTRACE = Y may generate very large logs and severely impact performance of the client. Do not set this unless instructed to do so by Technical Support.
ZTRACEL	**Trace level (000/040/999)** Specify tracing level. If blank, use existing value. **Caution:** Setting ZTRACEL to a higher number may generate very large logs and severely impact performance of the client. Do not set this unless instructed to do so by Technical Support.

Hardware Scan Options (RADHWCFG)

Use instances in the RADHWCFG class in the CLIENT domain to specify the type of hardware scans you want performed on the client device. Hardware scan information is reported in the ZCONFIG object. To implement the hardware scan options, connect an instance of the RADHWCFG class to an instance in the LOCATION class.

Note

Client Operations Profiles must be enabled to use the RADHWCFG class. For testing, you may want to create a RADHWCFG object on the client device with all the attributes in the RADHWCFG class. Then, change the attributes to Y or N to see the result in the ZCONFIG object.
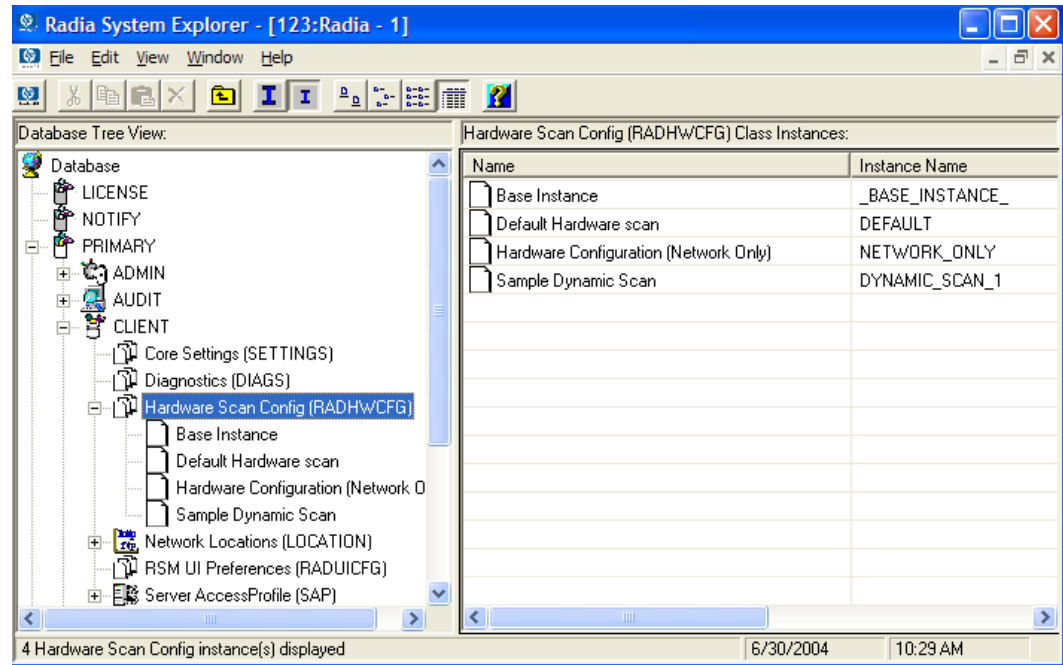
Figure 6.8 ~ View the sample instances in the RADHWCFG class.

We provide you with four sample instances in RADHWCFG.

■ **Base Instance**
Create copies of the base instance to create your own hardware scans.

■ **Default Hardware Scan**
This instance scans for the most commonly requested information.

■ **Hardware Configuration (Network Only)**
This instance scans for network information only.

■ **Sample Dynamic Scan**
This instance provides samples using the Dynamic Scan variables.

The table below details each of the possible hardware scans. Examples of the ZCONFIG attributes that may be returned are provided.

Note

Note that the attributes returned depend on what the hardware configuration is. For example, if the client device has only one printer connected, then only one ZHDWPA0n attribute will be reported in ZCONFIG.

Table 6.6 ~ Attributes in the RADHWCFG Class

Attribute	Description
NAME	**Friendly Name** The friendly name of the instance.
CPU	**CPU [Y/N]** Specify **Y** to scan for CPU information. **ZCONFIG attributes:** ZHDWBIOS, ZHDWCOMP, ZHDWCPU, ZHDWCPUN, ZHDWCPUS, ZHDWFPU, ZHDWXPAG, ZHWCPU01, ZHDFPU01
OS	**OS [Y/N]** Specify **Y** to scan for Operating System information. **ZCONFIG attributes:** REBOOTD, REBOOTT, WTSSRVR, ZHDWLANG, ZHDWOS, ZHDWOSDB, ZHDWOSOG, ZHDWOSOW, ZHDWSVCP
MEMORY	**Memory [Y/N]** Specify **Y** to scan for memory information. **ZCONFIG attributes:** ZHDWMEM, ZHDWMEMF
HDLOCAL	**Local Drives [Y/N]** Specify **Y** to scan for internal hard drives. **ZCONFIG attributes:** ZHDWCDDR, ZHDWD00, ZHDW00C, ZHDWD00F, ZHDWD00S, ZHDW00T, ZHDWD01, ZHDW01C, ZHDWDF_A, ZHDWDLST, ZHDWDNUM
HDREMOTE	**Remote Drives [Y/N]** Specify **Y** to scan for external hard drives. **ZCONFIG attributes:** ZHDW00, ZHDWD00C, ZHDWD00F, ZHDW00S, ZHDW00T, ZHDWDLST, ZHDWDNUM
NETWORK	**Network [Y/N]** Specify **Y** to scan for network information. **ZCONFIG attributes:** GATEWY01, IPADDR01, LADAPT01, NETLOC01, SUBNET01, ZGATEWAY, ZHDWIPAD, ZHDWLANA, ZHDWNET1, ZHDWNNET, ZNETLOC, ZSUBNET
PERIPHER	**Peripherals [Y/N]** Specify **Y** to scan for peripherals such as keyboard and mouse. **ZCONFIG attributes:** ZHDWKYBD, ZHDWMOUS, ZHDWPPAR, ZHDWPSER, ZHDWVIDO, ZHDWVRES

Table 6.6 ~ Attributes in the RADHWCFG Class	
Attribute	**Description**
PRINTER	**Printers [Y/N]** Specify **Y** to scan for printers. **ZCONFIG attributes:** ZHDWPA00, ZHDWPA01, ZHDWPPRN
HAL_VER	**HAL Statistics [Y/N]** Specify **Y** to scan for the HAL (Hardware Abstraction Layer) version. **ZCONFIG attributes:** HALCOMP, HALDATE, HALFNAME, HALFVER, HALINAME, HALLANG, HALPNAME, HALPVER, HALSIZE.
APP_VER	**Application Version [Y/N]** Specify **Y** to scan for versions of MSI (ZHDWVMSI) and IE (ZHDWVIE).
WMISCAN	**Use WMI to collect data [Y/N]** Specify **Y** to perform the scan using WMI (Windows Management Instrumentation).
DSCAN00n	**Dynamic Scan 00n** Specify **Y** to use the dynamic scan variable. See *Dynamic Scanning* below.

Dynamic Scanning

In addition to the built in scans, create your own scans using the Dynamic Scan (DSCAN00n) instances. There are three types of dynamic scan instances WMI, Registry and File. The format for a dynamic scan is: VariableName = Type(Parm1, Parm2, …) where VariableName is the attribute in ZCONFIG where you want the information to be reported, Type is WMI, Registry or File, and Parm$n$ is the query for the information. Three examples are provided below.

Example 1: **WMI**

A WMI scan would use the following format: VariableName = WMI(WQL Statement, Property, Default).To collect the Model of a client device using WMI, create a DSCAN000 variable similar to:

```
HWMODEL=WMI("Select * from Win32_ComputerSystem"; Model; NONE)
```

This scan would create the variable ZCONFIG.HWMODEL, and populate it with the client device's model.

Example 2: Registry

To scan a registry key to determine where Adobe 5.0 is installed, create a DSCAN001 variable similar to:

```
ADOBEPTH=REG(HKLM\SOFTWARE\Adobe\Acrobat_Reader\5.0\InstallPath)
```

The result will be reported in ZCONFIG.ADOBEPTH.

> **Note**
>
> When scanning for a "default" registry value, the path to the registry key must end with a backslash. For example, to read the default value of the Installer key type:
> ADOBEPTH=REG("HKLM\SOFTWARE\Adobe\Acrobat Reader\6.0\Installer\")
>
> To read the Path value of the Installer key, type:
> ADOBEPTH=REG("HKLM\SOFTWARE\Adobe\Acrobat Reader\6.0\Installer\Path")

Example 3: File

Dynamic File scanning can return size (SIZE), date stamp (DATE), file version (FVER), product version (PVER), and time (TIME) stamp of a specified file. You may request any combination of these properties. To scan for the file C:\temp\test.exe, create a DSCAN002 similar to:

```
TEST####=FILE(c:\Temp\Test.exe;SIZE,DATE,FVER,PVER,TIME)
```

The #### will be replaced by the corresponding file property name. One attribute will be created in the ZCONFIG object for each file property for which you scanned. In this example, five variables will be created based on the information collected on the C:\temp\test.exe file, ZCONFIG.TESTSIZE, ZCONFIG.TESTDATE, ZCONFIG.TESTFVER, ZCONFIG.TESTPVER, and ZCONFIG.TESTTIME.

Setting User Interface Properties (RADUICFG)

Use the RADUICFG class to specify settings for the Radia Software Manager User Interface. You must be licensed for the Radia Software Manager client. To implement the hardware scan options, connect an instance of the RADUICFG class to an instance in the LOCATION class.

> **Note**
>
> You must be licensed for and install the Radia Software Manager to use this class.
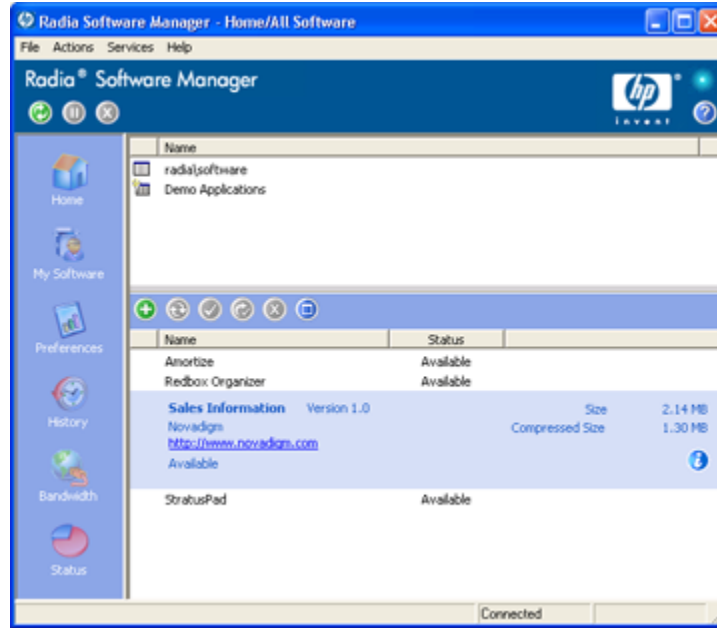
Figure 6.9 ~ View the Radia Software Manager User Interface.

Table 6.7 ~ Attributes of the RADUICFG Class

Attribute	Description
PNLOUTBR	**Display Outbar [Y/N]** Specify Y to display the Radia Bar. This is located on the left side of the panel and provides navigation throughout the interface.
BNHOME	**Display Home Button [Y/N]** Specify Y to display the Home button on the Radia Bar.
BNMYSOFT	**Display My Software [Y/N]** Specify Y to display the My Software button on the Radia Bar.
BNPREFER	**Display Preferences [Y/N]** Specify Y to display the Preferences button on the Radia Bar.
BNBNDWTH	**Display Bandwidth [Y/N]** Specify Y to display the Bandwidth button on the Radia Bar.
BNHISTRY	**Display History [Y/N]** Specify Y to display the History button on the Radia Bar.
BNSTATUS	**Display Status [Y/N]** Specify Y to display the Status button on the Radia Bar.

Table 6.7 ~ Attributes of the RADUICFG Class

Attribute	Description
SHWMENUS	**Show Menus [Y/N/U]**
	Specify U to allow the user to control appearance of the menu bar. Specify Y or N to turn the Menu bar on or off, and not allow the user to control its appearance.
SHWCATLG	**Show Catalog [Y/N/U]**
	Specify U to allow the user to control appearance of the list of catalogs. Specify Y or N to turn the catalog list on or off, and not allow the user to control its appearance.
STRTCHNG	**Startup param modifications [Y/N]**
	Specify Y to allow the user to modify the startup parameters in General Options in Preferences. Specify N to not allow the user to modify the startup parameters.
STRTFILE	**Startup parameter filename**
	Specify the filename for the startup parameters found in the General Options in Preferences.
STRUPMSG	**Warn if Startup file changes [Y/N]**
	Specify Y to warn the user if the startup parameter file has changed.
ASKOFFL	**Prompt for offline mode [Y/N/U]**
	Specify U to allow the user to control prompting for offline use of Radia Software Manager. Specify Y or N to turn the prompt on or off, and not allow the user to control the prompt.
BWSTRTUP	**Show BW Control on Startup [A/Y/N]**
	Set to A to automatically display the bandwidth control when processing a service that has bandwidth settings. Set to Y to always display whether or not the service has bandwidth settings and N never to display.
COLORSET	**Colors [SYSTEM/DEFAULT/CUSTOM/USER]**
	Select SYSTEM to use the operating system colors, DEFAULT to use the Radia default color scheme, CUSTOM to use COLORSEL, COLORBAK, COLORBTN and COLORWK. The user will not be able to change the colors if SYSTEM, DEFAULT, or CUSTOM is selected. Select USER to allow the user to control the colors.
COLORSEL	**Specify RGB or actual Color Select**
	Specify color for selection areas. Check the Microsoft Web site for possible colors.
COLORBAK	**Specify RGB or actual Color Back**
	Specify color for the interface's background. Check the Microsoft Web site for possible colors.
COLORBTN	**Specify RGB or actual Color Button**
	Specify the button colors. Check the Microsoft Web site for possible colors.
COLORWK	**Specify RGB or actual Color WorkAr**
	Specify the color for the work area. Check the Microsoft Web site for possible colors.
STATSTRT	**Status window on startup [Y/N]**
	Set to Y to show the status window on start up.

Table 6.7 ~ Attributes of the RADUICFG Class

Attribute	Description
CUSTIMG	**Custom Image File or RSM Banner** Specify a custom image file or banner. Acceptable file types are JPG/JPEG, GIF, TIF, and BMP. The size limitations in pixels are approximately height of 60 and a width of 250. If no location is specified for the file, the default is IDMLIB (<System Drive:>\Program Files\Novadigm\Lib).
CUSTURL	**Custom URL on RSM Banner** Specify a URL that the client computer's default Internet browser will open to if the subscriber clicks on the CUSTIMG.
CUSTTEXT	**Custom Hover Text on RSM Banner** Specify the text to display when the client computer's mouse hovers over the CUSTOMIMAGE.
CUSTTTLE	**Custom Title on RSM Banner** Specify the text to display in the Radia Software Manager's title bar.
COLTYPE	**Columns [Forced/Required]** Set to **Forced** if you want only the columns specified in COLNAMES to appear. Set to **Required** if at least the columns specified in COLNAMES should appear. Name and Status are always displayed.
COLNAMES	**Name of Columns (CSV format)** Specify the columns you want displayed. Separate the columns with a comma.
EXPSITEM	**Expand Active Service item [Y/N/U]** Specify U to allow the user to control the expansion of the active Service List item. Specify Y or N to allow or not allow the user to expand the active item in the Service List.
EXPCITEM	**Expand Active Catalog Item [Y/N/U]** Specify U to allow the user to control the expansion of the active catalog item. Specify Y or N to allow or not allow the user to expand the active catalog item.
SHWGRID	**Show Grid Lines [Y/N/U]** Specify U to allow the user to control the display of grid lines. Specify Y or N to turn off or on the display of grid lines.
SHWADVOP	**Show Advanced Options [Y/N/U]** Specify U to allow the user to control the display of Advanced Options. Specify Y or N to turn off or on the display of Advanced Options such as Download Only, Reconfigure, and Undo buttons.
PROXYUSE	**Use Proxy Server [Y/N/U]** Specify U to allow the user to control the use of an internet proxy. Specify Y or N to turn the use of an internet proxy on or off, and not allow the user to control the use of an internet proxy.
PROXYDSC	**Discover proxy address [Y/N/U]** Specify U to allow the user to control internet proxy discovery. Specify Y or N proxy discovery on or off, and not allow the user to control its appearance.
PROXYADD	**Proxy Server Address** Specify the internet proxy server's address.

Table 6.7 ~ Attributes of the RADUICFG Class

Attribute	Description
PROXYPRT	**Proxy server Port** Specify the internet proxy server's port.
BTNINST	**Enable Install Button [Y/N]** Specify Y to enable the Install button.
BTNUPDT	**Enable Update Button [Y/N]** Specify Y to enable the Update button.
BTNDWLD	**Enable Download Button [Y/N]** Specify Y to enable the Download button.
BTNRECFG	**Enable Reconfigure Button [Y/N]** Specify Y to enable the Reconfigure button.
BTNUNDO	**Enable Undo Button [Y/N**] Specify Y to enable the Undo button.
BTNVRFY	**Enable Verify Button [Y/N]** Specify Y to enable the Verify button.
BTNREPR	**Enable Repair Button [Y/N]** Specify Y to enable the Repair button.
BTNDEL	**Enable Delete Button [Y/N]** Specify Y to enable the Delete button.
BTNCANCL	**Enable Cancel Button [Y/N]** Specify Y to enable the Cancel button.
BTNPAUSE	**Enable Pause Button [Y/N]** Specify Y to enable the Pause button.
SHWCOLEX	**Show Expand/Collapse Button [Y/N]** Specify Y to show the Install button .
SHWINFO	**Show/Hide Extended Info [Y/N]** Specify Y to show the Extended Info button when a service item is expanded.
SHWSCHEV	**Show/Hide Scheduled Event [Y/N]** Specify Y to show the Scheduled Event button when a service item is expanded. This button looks like a clock.
TMNUTXT0n	**RadTray Menu Text n** Create a custom menu item for the Radia System Tray. This menu will be available when you right click on the Radia System Tray icon. To create a separator bar, type SEPARATOR as the menu text.
TCMDTXT0n	**RadTray Menu Command n** Create a custom menu item for the Radia System Tray. Specify a command to run when TMNUTXTn is clicked in the Radia System Tray. The command must be available from the IDMSYS directory.
NAME	**Friendly Name** Friendly name of the instance.

Client Operations Profile Example

This section provides a simple example of how to configure a Client Operations Profile. The driving force is to have your client computers connect with the most appropriate Radia Configuration Server. Usually, you will want to assign your client computers to a Radia Configuration Server based on network address.

Scenario

Suppose you divide your enterprise into two regions, EAST and WEST. All client computers in the EAST region are in the 192.111.111.0 network, and all client computers in the WEST region are in the 193.111.111.0 network. In addition, suppose you have two Radia Configuration Servers, one called RCS_EAST as the primary Radia server for the EAST region, and one called Radia Configuration RCS_WEST as the primary Radia server for the WEST region.

To configure the sample scenario

1. Build two Server Access Profile (SAP) instances, one for RCS_EAST and one for RCS_WEST. *Figure 6.10 ~ View Sample_RCS_East* below shows a SAMPLE_RCS_EAST.
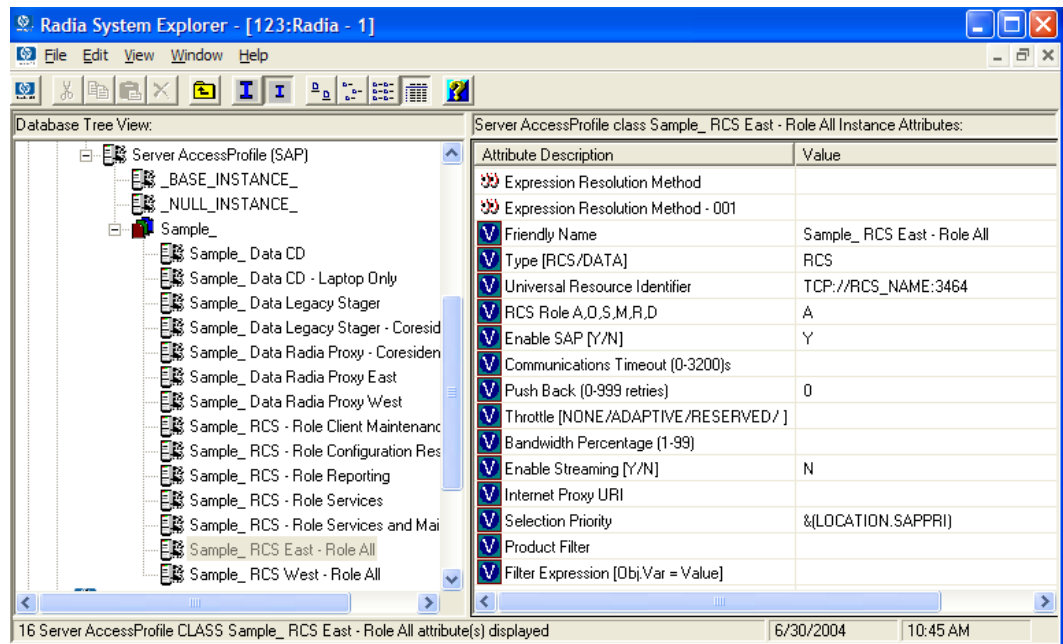
Figure 6.10 ~ View Sample_RCS_East.

In the Server Access Profile (SAP) class use the TYPE attribute to specify a server as type DATA or RCS. In this example, we will be configuring only Radia Configuration Servers. Therefore, all servers will have SAP.TYPE set to RCS.

For each Server Access Profile instance, you must also identify a role. Again, for simplicity, we will set SAP.ROLE to A for all. This means that the Radia Configuration Server can provide client operations profiles, service resolution, maintenance, data, and reporting.

At a minimum, you will need to specify the Universal Resource Identifier (URI) attribute. Customize other variables as needed.

2. Build two location instances, one for the EAST region, and one for the WEST region.

Create a location instance called 192_111_111_0 with a friendly name of Sample_Location East, and a location instance called 193_111_111_0 with a friendly name of Sample_Location West. See the figure below for an example.
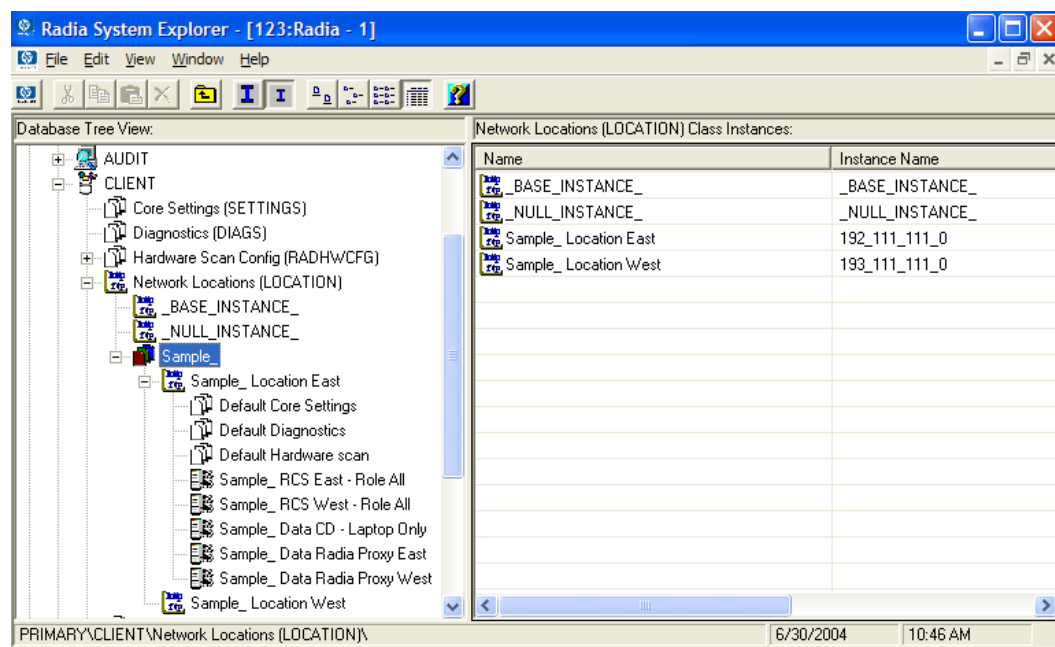
Figure 6.11 ~ Connect location instances.

3. Connect the LOCATION instance to the appropriate Server Access Profile (SAP) instance.

- In the LOCATION.Sample_Location East instance define a connection to the SAP.Sample_RCS EAST.

- In the LOCATION.Sample_Location West instance define a connection to the SAP.Sample_RCS WEST.

4. Now, consider what you want to happen in the following cases:

   - Suppose you are a client in the EAST Region and the RCS_EAST is unavailable. Your options are:

     Abort

     Or

     Go to RCS_WEST as a second choice.

   - Suppose you are a client in the EAST region and the RCS_EAST is busy. In other words, the task limit defined in your Radia Configuration Server settings file has been reached. Your options are:

     ♦ Continue to retry the RCS_EAST until a connection is available

     Or

     ♦ Go to RCS_WEST as a second choice.

Once you understand the concepts involved, and feel comfortable with this process, you can begin to add other components to the Client Operations Profile. After you finish with TYPE=RCS, configure your servers with TYPE=DATA. In addition, you can identify particular servers of TYPE=RCS to use different ROLES.

Summary

■ Use Client Operations Profiles to provide redundancy in your environment.

■ Select which servers will perform which roles.

■ You can assign client computers to specific servers based on network location or any other criteria.

■ You must enable Client Operations Profiles in the Radia Database and on the client computer.

7

Preparing Services

At the end of this chapter, you will:

- ■ Be aware of service options.
- ■ Know how to create a service from a promoted package.
- ■ Know how to restart the client computer.
- ■ Know how to install services under the system account.
- ■ Be familiar with preparing versioned applications.

This guide covers the *suggested* implementation for the Radia Application Manager. Although you will tailor this strategy to meet your organization's needs, it is recommended that you review this guide for a comprehensive understanding of the Radia Application Manager. This chapter covers preparing services.
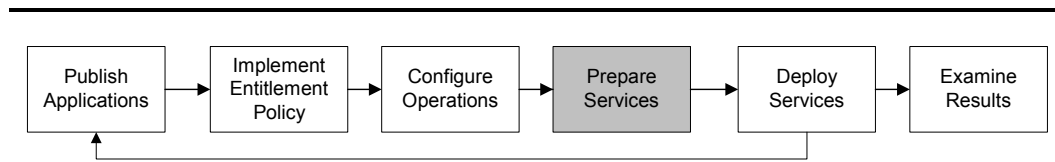
Figure 7.1 ~ Tasks completed in this guide.

Configuring Applications in the Machine/User Context

Your enterprise may require that applications are configured to accommodate multiple users or that one computer always have the same applications available without regard to the current user. A service may be considered multi-context if it has components to be installed in the machine context and in the user context.

> **Note**
>
> This feature is applicable to Radia Application Manager clients only.

In order to complete the installation of a multi-context service, the client computer will need to connect to the Radia Configuration Server twice; once to install the machine components, and once to install the user components. The machine components are installed first. If the machine portion does not complete successfully, installation of the user component will not take place. The machine portion can be invoked through notify or timers. The user portion should be embedded in a logon script, desktop shortcut, or batch file that is local to the user.

Complete the following steps to configure your service for machine or user installations:

1. Use Radia Publisher to specify whether a package component is to be installed in the User or Machine context, or use Radia System Explorer to specify a component's context.

2. Use Radia System Explorer to configure your service to deploy in either the system or user context.

3. Use Radia System Explorer to set the deployment methods for the service.

Setting the Context of Components (ZCONTEXT)

Use the Radia Publisher to select machine and user components separately. Set the context of a component either using the Radia Publisher during the publishing process, or using the Radia System Explorer after the publishing process.

> **Note**
>
> To take advantage of the Machine/User context, previously packaged applications must be repackaged.

To set the context of a component in Radia Publisher

1. Right-click the files or directories and select **Set Properties**.

   The **Instance Properties** dialog box opens.

2. In the **Client Management** tab, select the appropriate context for the component as shown in Figure 7.1 on page 302 and described in Table 7.1 on page 304.
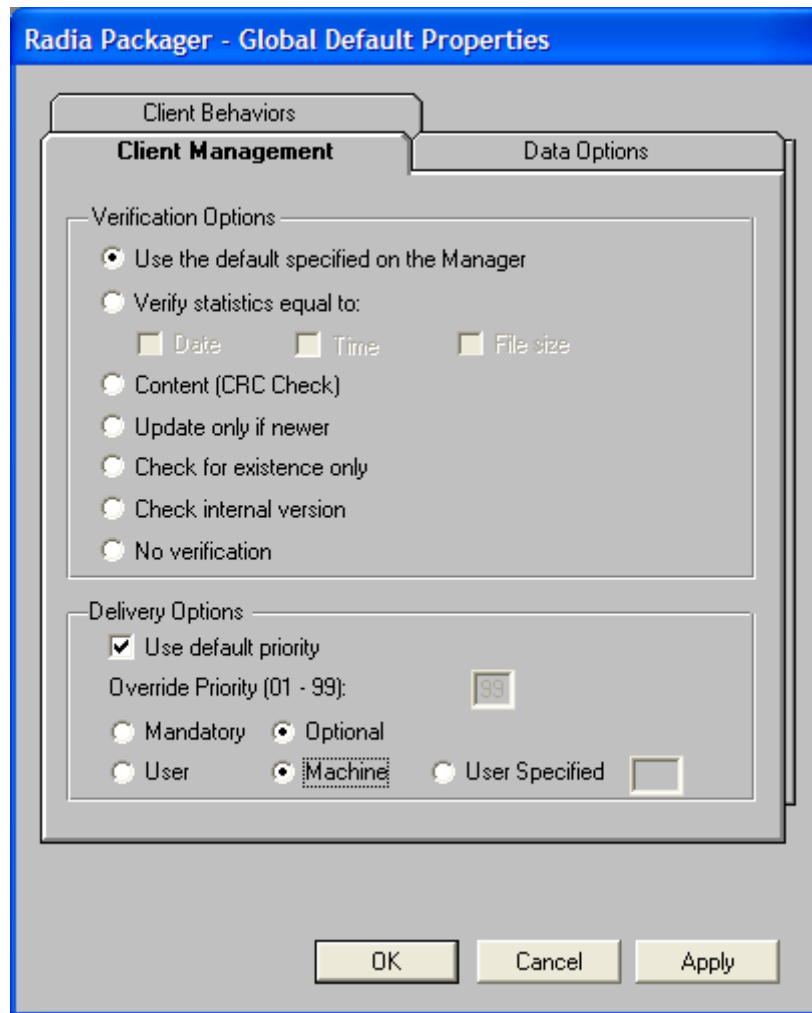
Figure 7.1 ~ Instance Properties - Client Management tab.

If you have already completed the publishing process, use the Radia System Explorer to set the ZCONTEXT attribute in the component's instance.

To set the context in Radia System Explorer

1. From the Start menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**.

The **Radia System Explorer Security Information** dialog box opens.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

   The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **SOFTWARE**.

5. Double-click **Application Packages (PACKAGE)**.

6. Double-click the component's class.

7. Double-click the component.

8. Double-click **ZCONTEXT**.
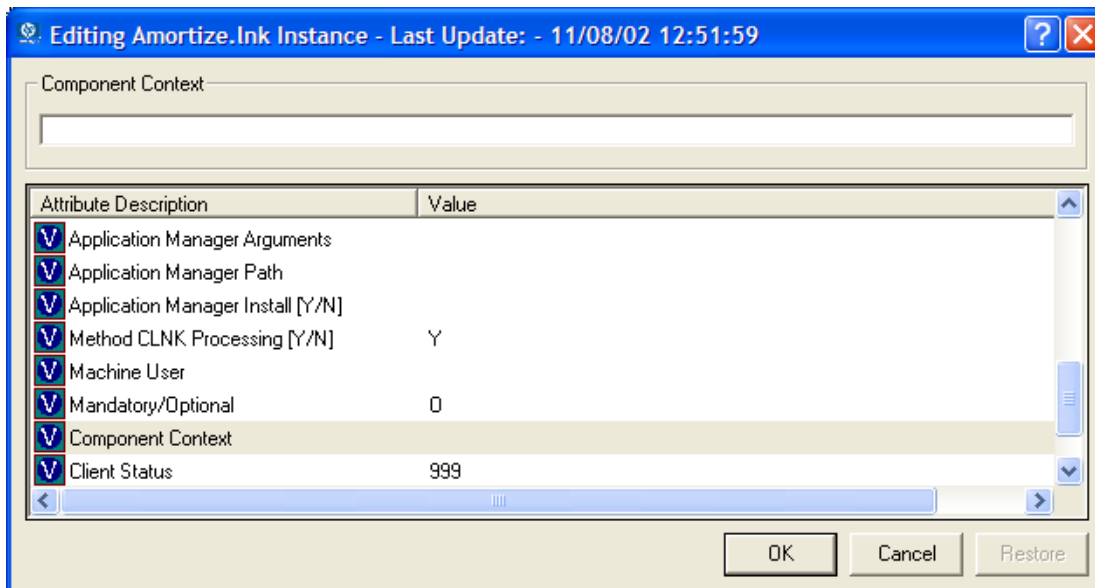
Figure 7.2 ~ Editing the ZCONTEXT attribute.

9. Type the appropriate value based on Table 7.1 on page 304.

Table 7.1 ~ Component Context (ZCONTEXT) Options	
Setting	**Explanation**
blank	Leave ZCONTEXT blank if the component is independent of context. This component will be installed during both machine and user connects.
U	Type U if you want to indicate that this component will be deployed only to the subscriber logged on when the application is initially deployed.
M	Type M to indicate that the file should be deployed to all users of the computer.
User Specified	*This option is for future use.*

10. Click **OK** to complete the changes.

11. Click **Yes** to confirm the changes and return to the Radia System Explorer.

> **Note**
>
> As a rule, the component will be processed if its ZCONTEXT attribute matches the value of the context parameter in the radskman command line or if the component's ZCONTEXT attribute is blank.

Now that you have set the component's context, you must specify that the ZSERVICE instance for the application has both machine and user components.

Setting the Service Mode (ZSVCMODE)

The ZSVCMODE attribute in the ZSERVICE class is used to determine if the machine/user context is relevant to the package's deployment. If you need to create the ZSVCMODE attribute, add it to the class template for the ZSERVICE class. Create it as a variable of length 3. Accept the other default properties. The possible values are **M**, **U**, **MU, blank,** and **EMU**.

> **Caution**
>
> We recommend that you back up the Radia Database prior to making class template changes.

Table 7.2 ~ Values for ZSVCMODE in the ZSERVICE Class	
Value	**Explanation**
Blank	Use this when you want the components to install whether the client is logged on in the machine or user context. The application will only be installed using the available context.

Table 7.2 ~ Values for ZSVCMODE in the ZSERVICE Class

Value	Explanation
EMU	Enhanced Machine/User service:
	If the Client Connect is being made in the user context, but the machine side of the application has not yet been installed, this will force the machine connect. After the machine connect completes successfully, the user connect is initiated to install the user components.
	Use this for optional applications that the user controls through the Radia Software Manager.
EMU:AD=N	Enhanced Machine/User service:
	If the Client Connect is being made in the user context, but the machine side of the application has not yet been installed, this will force the machine connect. After the machine connect completes, the user connect is initiated to install the user components.
	Use this for optional applications on a shared computer that the user controls through the Radia Software Manager. The addition of AD=N prevents a user from being able to remove the machine components of the application. Other users of the client computer may still need the machine components of the application. The possible application events are:
	AI = application install
	AD = application delete
	AU = application update
	AR = application repair
	VA = version activation
	VD = version deactivation
	The default for each event is Y. Separate multiple events with a comma.
M	Machine service only
	Set ZSVCMODE to M if the service has only machine components. This service will be ignored if context is set to u on the radskman command line.
MU	Machine/User service
	Set ZSVCMODE to MU if the service has both machine and user components. The user connect will verify that the machine components have been installed before installing the user components. The user components will not be installed if the machine components are not present.

To set ZSVCMODE in the ZSERVICE

1. From the **Start** menu, select **Programs, Radia Administrator Workstation, Radia System Explorer**.

The **Radia System Explorer Security Information** dialog box opens.

> ### Note
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

3. The Radia System Explorer window opens.

4. Double-click **PRIMARY**.

5. Double-click **SOFTWARE**.

6. Double-click **Application (ZSERVICE)**.

7. Double-click the appropriate service.

8. Double-click **ZSVCMODE** in the list view.

9. Type the appropriate values based on the values shown in Table 7.2 on page 304.

10. Click **OK** to complete the changes.

11. Click **Yes** to confirm the changes and return to the Radia System Explorer.

Enhanced Machine User Services for Radia Software Manager

When an application has machine and user components, the Radia Software Manager client needs elevated privileges on the client device to install machine components properly, and the user components need a user logged on to access the user's settings. A machine component might be a file or registry key while a user component might be a desktop shortcut. If context is set to U the subscriber will not be able to install the machine side of the application, and the user connect will fail. By setting ZSVCMODE to EMU, when the subscriber selects the application to install, the client will check to see if the machine components are already installed. If they are not, Radia Software Manager will install the machine components *for that service only*, and, then, install the user components. In other words, two separate Client Connects will run, one in machine context and one in user context. Set the CONTEXT tag to U in the args.xml file to enforce EMU behaviors.

Note
ZSVCMODE must be changed to 3 bytes and EMU must be one of the choices.

If ZSERVICE.ZSVCMODE is set to EMU, when the Radia Software Manager client encounters a service that is marked as EMU it first checks to see if the machine catalog exists, and if the requested service exists in the machine catalog:

■ If the service exists, and is marked installed in the machine catalog, the service is installed for the user.

■ If the service is not installed on the machine side, but the catalogs are synchronized, then the client will first install the machine side of the service and then install the user piece of the service.

■ If the machine catalog is missing (because a machine connect never ran), then the client gets the machine catalog, and installs the service on the machine side. Then, if the service was successfully installed on the machine side, the user side of the service will be installed.

Deploying a Machine/User Service

The client will need to connect to the Radia Configuration Server twice to complete the installation of services that are marked with ZSVCMODE set to MU. The first connect will be in the machine context. In the **radskman** command line, add a parameter of context=M. This will set the startdir=SYSTEM by default. Do this by using either a Timer or a Notify command. For information on how to create a **radskman** command line see *Specifying the Command Line* on page 365.

The second connection will install the user components. Do this in either a logon script, batch file, or desktop icon since the user needs to be logged on. Use the **radskman** command line with the context=U parameter added. If ZSVCMODE is MU, the user components will *only* be installed if the machine connect has finished successfully.

Radia Service Groups

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

This includes MSI packaged products where:

■ A product may utilize more than one MSI service-package, for example, where MS Office may require other language pack services.

■ A large product, MS Office for example, may need to be split into smaller sub-services to install only specific parts of the product suite.

The Radia Software Manager user interface will display only the master service.

For detailed information on creating Radia Service Groups, refer to the *Radia System Explorer Guide.*

Installing Services with the System Account (ZSYSACCT)

For computers running Windows NT, Windows 2000, or Windows XP, you can specify whether to install the service under the system account or the user's account. To do this, modify the ZSYSACCT attribute in the Application (ZSERVICE) instance. This attribute controls whether to use the system account for installation on a per service basis.

Table 7.3 ~ ZSERVICE Attribute: ZSYSACCT

Attribute	Description
ZSYSACCT	Set to Y to install the application under the system rights. The type of connect is ignored.
	Set to N or blank to install the application under the user who is currently logged on. The type of connect is ignored.
	Set to M to install the application under the System Account if the current connect is in the machine context (context=m).
	Set to U to install the application under the System Account if the current connect is in the user context (context=u).
	Default: N

To edit the ZSYSACCT attribute in the Application instance

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

The **Radia System Explorer Security Information** dialog box opens.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, enter a **User ID** and **Password**, and then click **OK**.

The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **SOFTWARE**.

5. Double-click **Application (ZSERVICE)**.

6. Double-click the appropriate application instance, such as **Amortize**.

7. The attributes appear in the list view.
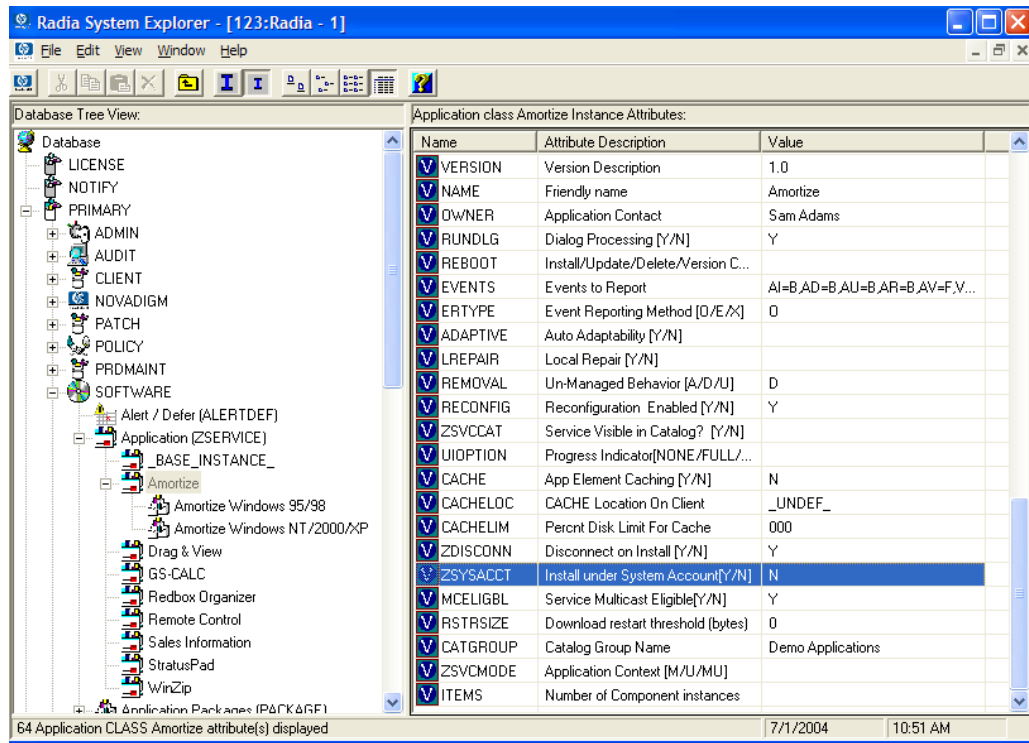
Figure 7.3 ~ Amortize instances, ZSYSACCT attribute.

8. Double-click **ZSYSACCT**.

The **Editing Instance** dialog box opens.
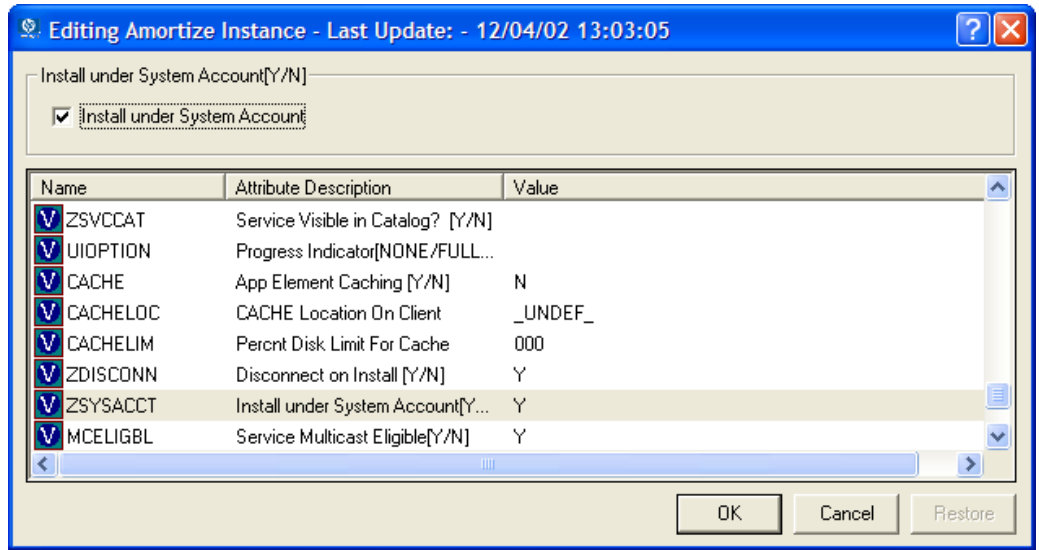
Figure 7.4 ~ Edit Instance dialog box with ZSYSACCT attribute selected.

9. Select **Install under System Account** to use System rights, or clear the check box to install under the rights of the user who is logged on.

10. Click **OK** to close the **Editing Instance** dialog box.

11. Click **Yes** to confirm your changes.

Local Catalog Processing

Local catalog processing reduces the network bandwidth and the number of connects required to the Radia Configuration Server to manage applications. Use a Timer or a Notify command to make the machine connect with context = m on the **radskman** command line.

Note
The ZSVCMODE attribute of the application must be M or MU to use this feature.

If you have only one user for a computer or multiple users on one computer with the same entitlements, local catalog processing allows you to make one connection to the Radia Configuration Server for both the machine and user components of a service. During the machine

connect, the required configuration information will be downloaded to the client computer, machine and user components of the services will be downloaded in a compressed format, and the machine components will be installed. For the user connect, set cat = m, local = y, and context = u on the **radskman** command line. When the user connect is made, the user's services will be resolved based on the machine's service list, and the already downloaded resources will be added, modified, or deleted as needed.

If a client computer has two or more users with distinct entitlements, then you will need to create policies for each machine so that all components for all of the services for all users of that client computer will be downloaded. Contact Professional Services for implementation.

Restarting the Client Computer

You may need to restart a client computer based on an application event. To do this, specify a reboot type and reboot modifiers in the ZSERVICE.REBOOT attribute. The modifiers allow you to:

■ set the type of warning message

■ handle a reboot with either a machine or user connect

■ and cause an immediate restart after the application event.

Caution

If the hreboot parameter is missing from the **radskman** command line, the parameter defaults to **Y** to handle service reboot requests. If you set hreboot to **p**, the client computer will *power down*, regardless of whether or not there is a service requiring a reboot.

First, specify the application event that needs the reboot. *Table 7.4 ~ Reboot Events and Codes* lists the codes for all possible application events. Set the application event code to a reboot type and any reboot modifier that you need to use. The sections below describe each type of reboot and all reboot modifiers.

If you need an application to immediately perform a hard reboot with no warning messages on application installation and repair, set the ZSERVICE.REBOOT variable to AI=HQI, AR=HQI.

Note

The parameters for the reboot attribute are not case-sensitive.

Table 7.4 ~ Reboot Events and Codes ~		
Application Events	**Code**	**Description**
Install	AI	Use AI to specify a reboot behavior for application installations. The default is no reboot.
Deinstall	AD	Use AD to specify a reboot behavior for application removals. The default is no reboot.
Locked File	AL	Use AL to specify a reboot behavior when a locked file is encountered. The default behavior when a locked file is encountered is to perform a Hard reboot with just an OK button (HY).
Update	AU	Use AU to specify a reboot behavior for application updates. The default is no reboot.
Repair	AR	Use AR to specify a reboot behavior for application repairs. The default is no reboot.
Version Activation	VA	Use AI to specify a reboot behavior for application version activations. The default is no reboot.

Reboot Types

After deciding which application events need a computer reboot, you will need to choose the type of reboot. Radia sends a message to the operating system that the computer needs to reboot. There are three types of reboot.

■ **Hard Reboot (H)**
All applications are shut down regardless of whether there are open, unsaved files or not. The subscriber will not be prompted to save open, modified files.

■ **Soft Reboot (S)**
Users are prompted to save their data if applications have open, unsaved files. If applications have unsaved data, the reboot will wait for the user to respond to the application's request for the user to save his data.

■ **No Reboot (N) (default reboot type)**
The computer will not restart after completing the specified application event. This is the default reboot type for all application events except a Locked File Event (AL). If you specify AL=N, then the client computer will not perform a hard reboot with an **OK** and **Cancel** button when a locked file is encountered. If no restart type is specified for an application event, no restart will occur.

Reboot Modifier: Type of Warning Message

You can specify the type of warning message you want to send to the subscriber before the restart occurs. If you specify a type of reboot, but do not specify a type of warning message, the default warning message for that type will be displayed. There are three types of warning messages.

Warning messages are displayed automatically for the Radia Software Manager and for Radia Application Manager used with the Radia System Tray. If you do not want to show a warning message, specify ask=N in a **radskman** command line.

■ **Quiet (Q)**
No reboot panel will be displayed.

■ **OK Button (A)**
A warning message will display with an **OK** button only. Clicking **OK** will initiate the reboot. The user will not be able to cancel the restart.

■ **OK and Cancel Button (Y)**
Clicking **OK** will initiate reboot. If the subscriber clicks **Cancel**, the reboot will be aborted.

> **Note**
>
> You can specify a timeout value for the Warning Message box by adding the RTIMEOUT value to the **radskman** command line. Set RTIMEOUT to the number of seconds you want the Radia Client to wait before continuing with the reboot process.

Reboot Modifier: Machine and User Options

The Radia Client can connect as a machine or as a user by specifying the context parameter on the **radskman** command line. Use the Machine/User reboot modifier to specify if the reboot should complete based on the type of connect.

■ **Reboot on Machine connect (blank)**
When a machine/user reboot modifier is not supplied, the default behavior will be to reboot only on a machine connect where context=m in **radskman**, or if the context parameter is not specified. This default behavior should satisfy the majority of reboot requirements.

■ **Reboot on User connect only (U)**
The reboot will be honored on a user connect only where context=u in **radskman** or if the context parameter is not specified. The reboot will NOT occur where context=m in **radskman**.

■ **Reboot on both Machine and User connect (MU)**
Reboot will only occur when both the machine and user components of the application are installed.

Reboot Modifier: Immediate Restart

You can modify each type of reboot by adding **I** for **Immediate**. Use Immediate when you want the computer to restart immediately after resolving the current service. Radia will resolve the rest of the subscriber's services after the computer restarts. If you specify **I**, but do not specify **H** or **S** as the type of reboot, a hard reboot will be performed.

Specifying Multiple Reboot Events

If you have two services that require a reboot event on the same Client Connect, the most restrictive reboot type and reboot panel will be used. The least restrictive reboot type is No Reboot (N), followed by Soft Reboot (S), and the most restrictive is Hard Reboot (H). The least restrictive reboot warning message supplies both an **OK** and **Cancel** button (Y), followed by an **OK** button only (A), and the most restrictive is completely quiet (Q).

Suppose a subscriber is assigned an application that needs a soft reboot with just an **OK** button on installation, AI=SA. The subscriber is also assigned a second application that needs a hard reboot that displays both an **OK** and **Cancel** button, AI=HY. After all of the subscriber's application events are completed, a Hard Reboot (H) with only an **OK** button displayed (A) will be performed.

Preparing Versioned Applications

Normally, when you deploy an application to a client computer, it is activated immediately. This is the case when you use Radia Scheduler or Radia Notify. However, you can use Version Groups to roll out a new version of an application to subscribers, and activate it upon delivery or at a pre-determined time. If the installation of the new version fails, Radia will automatically roll back to the previous version. If problems occur in the new version after installation, you can deactivate the new version and roll back to the previous version for some, or all, subscribers.

After versioning is configured, the compressed files are stored on the client computer, and the versioning action takes place. The roll forward/roll backward activity can be entirely local, not requiring any data to be transferred at the version change time. It can also be configured to be partially local, with a minimum of data transmitted.

Versioned vs. Non-Versioned Applications

Versioned and non-versioned applications adhere to different connection models within the Radia Database. For non-versioned applications, one application instance connects to one or more package instances.
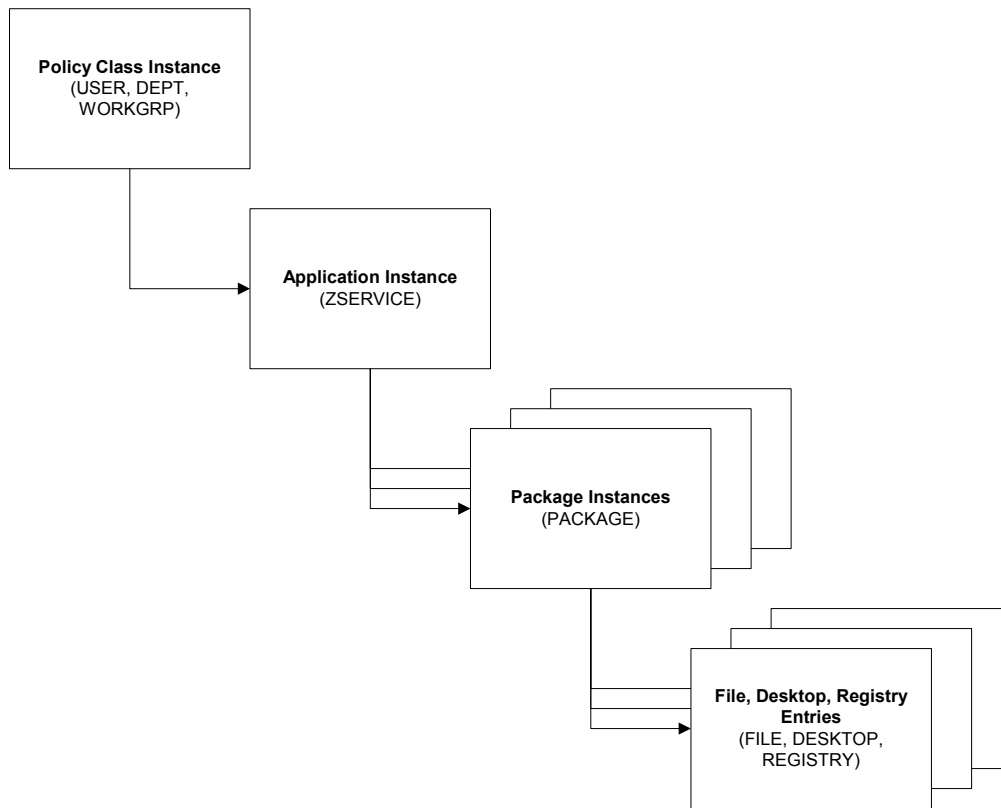
Figure 7.5 ~ Model for non-versioned deployments.

Versioned applications adhere to a different connection model than non-versioned applications. For versioned applications, an Application instance (ZSERVICE) connects to a single Version Group (VGROUP) instance.

> **Note**
>
> If you want to use multiple Version Groups, you must create one Service for each Version Group.

The Version Group instance connects to one or more Version instances that connect to one or more Package instances. A Version instance (which represents one version of a software application) contains one Radia package. Each Radia package is represented in the Radia Database by an instance of the PACKAGE class.
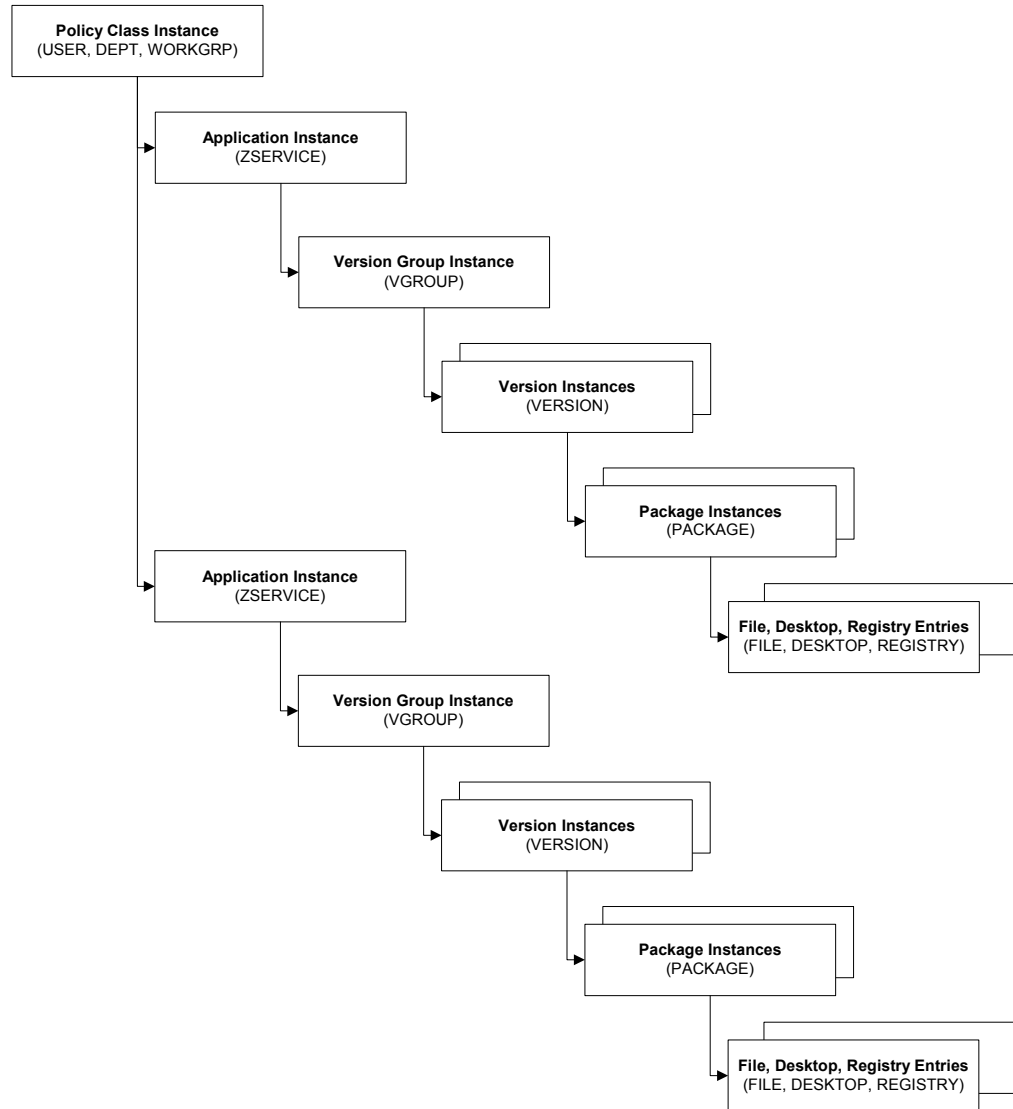
Figure 7.6 ~ Model for versioned deployment.

To prepare versioned applications

1. Use the **Radia Publisher** to package the application.

2. Right-click the **ZSERVICE** class and select **New Instance**.

3. In the **Create Instance** dialog box, type a **display name** and an **instance name**.

4. Click **OK**.

See the *Version Group Editor* topic to finish creating the Version Group.

The Version Group Editor

Use the Version Group Editor, in the Radia System Explorer, to create, edit, or delete instances for each version of an application, as well as manage the deployment of a *version group*. A version group contains all of the versions of an application.

To manage a versioned application, create an instance in the Version Group class, which represents the set of versions for the application. Then, use the Version Group Editor to create instances for each version of the application.

Creating a Version Group

In the following example, we will use the Radia System Explorer to create a new instance in the Version Group (VGROUP) class.

To create a Version Group

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

The **Radia System Explorer Security Information** dialog box opens.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **SOFTWARE**.

5. Right-click **Version Group (VGROUP)**.

6. Select **New Instance**. The **Create Instance** dialog box opens.

7. Type a name for the Version Group in the text field in the **Create Instance** dialog box, such as **Amortize**.

Figure 7.7 ~ The Create Instance dialog box.

8. Click **OK**.

The **Editing Version Group** dialog box opens.
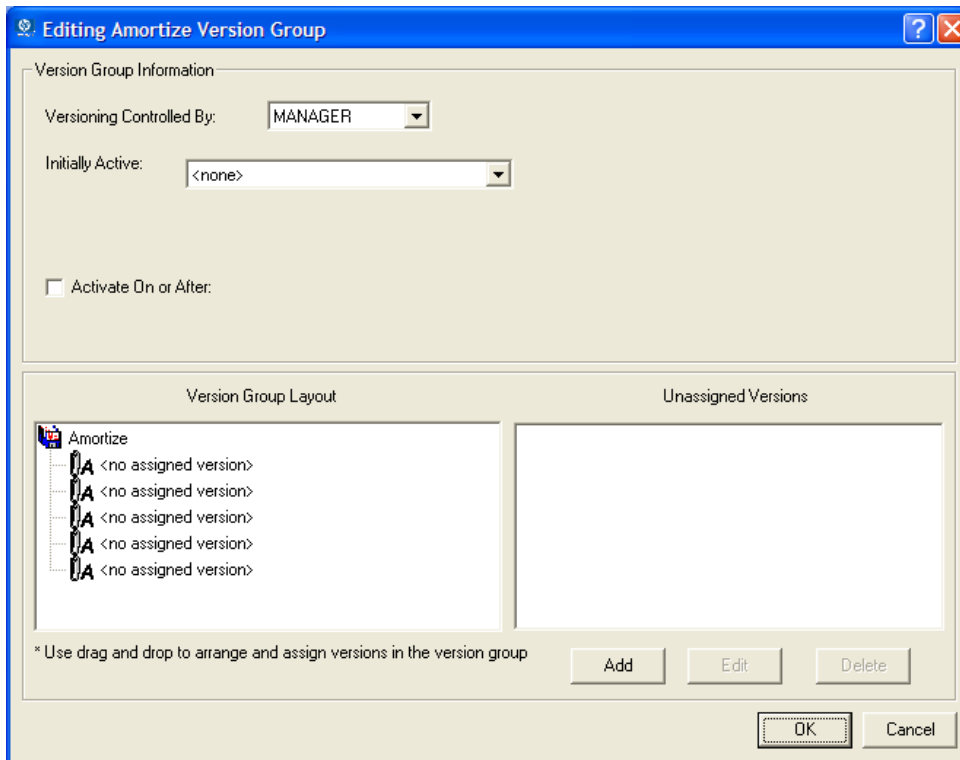
Figure 7.8 ~ The Editing Version Group dialog box.

Creating a Version Instance

Now that you have created a Version Group (VGROUP) instance, you will learn how to create an instance for each version of your application.

To create a version instance

1. In the Version Group Editor, click **Add**.

The **Create Version** dialog box opens.

2. Type a suffix that identifies the version. For example, type **Version 1**. The Version instance will be named Amortize Version 1.
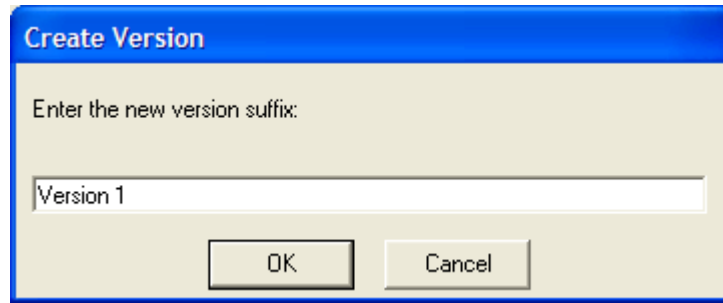
Figure 7.9 ~ Create Version dialog box.

3. Click **OK**.

   The **Version Editor** dialog box opens.
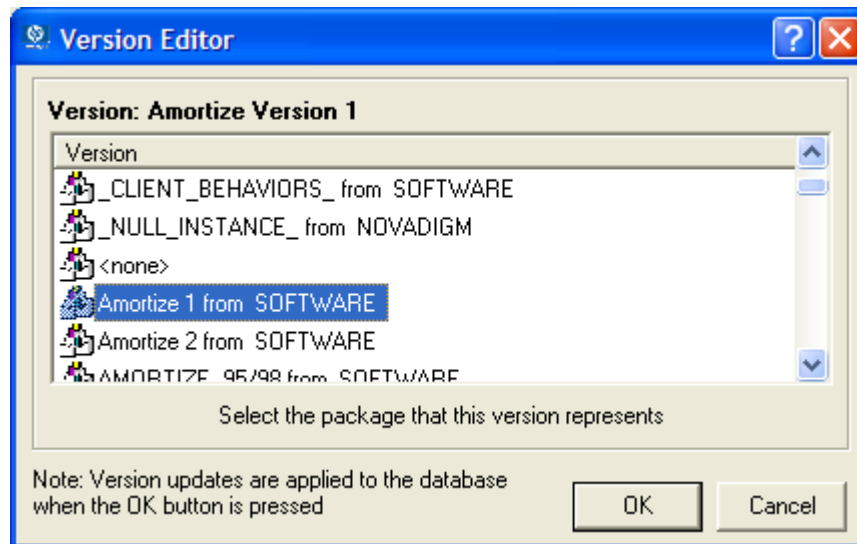
Figure 7.10 ~ Version Editor dialog box.

The Version Editor dialog box contains a list of Application Package (PACKAGE) instances stored in the Radia Database. Use this dialog box to connect the new Version (VERSION) instance to an Application Package (PACKAGE) instance. There is a one-to-one correspondence between these two instances.

4. Click the appropriate Application Package (PACKAGE) instance, such as **Amortize1**.

5. Click **OK**.

The Version instance appears in the **Unassigned Version** list.
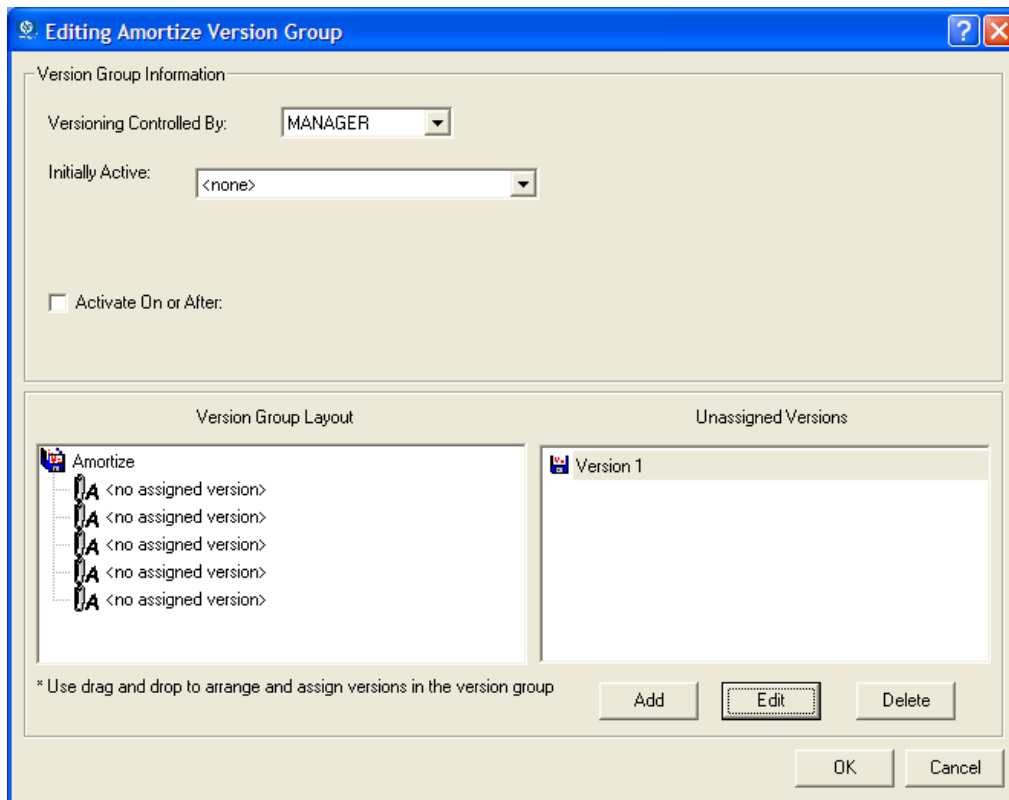
Figure 7.11 ~ Version 1 instance.

Add a Version instance for each version of the application that will be available to subscribers through this Version Group.

To delete a version instance

1. In the **Version Group Layout** list, select the version that you want to delete.

2. Click **Delete**.

The instance for the version appears in the **Unassigned Versions** list. The icon is in a dimmed state, ready for deletion.

To restore the instance, click the instance in the **Unassigned Versions** list, and then click **Un-delete**.

The version instance will not be deleted until you close the **Version Group Editor**.

3. Click **OK** to close the Version Group Editor.

Assigning Version Instances to the Version Group

After creating your Version instances, you must assign them to the Version Group.

To assign Version instances to the Version Group

■ In the **Unassigned Versions** list, click a Version instance and drag it over a connection labeled **<no assigned version>** in the Version Group Layout list.
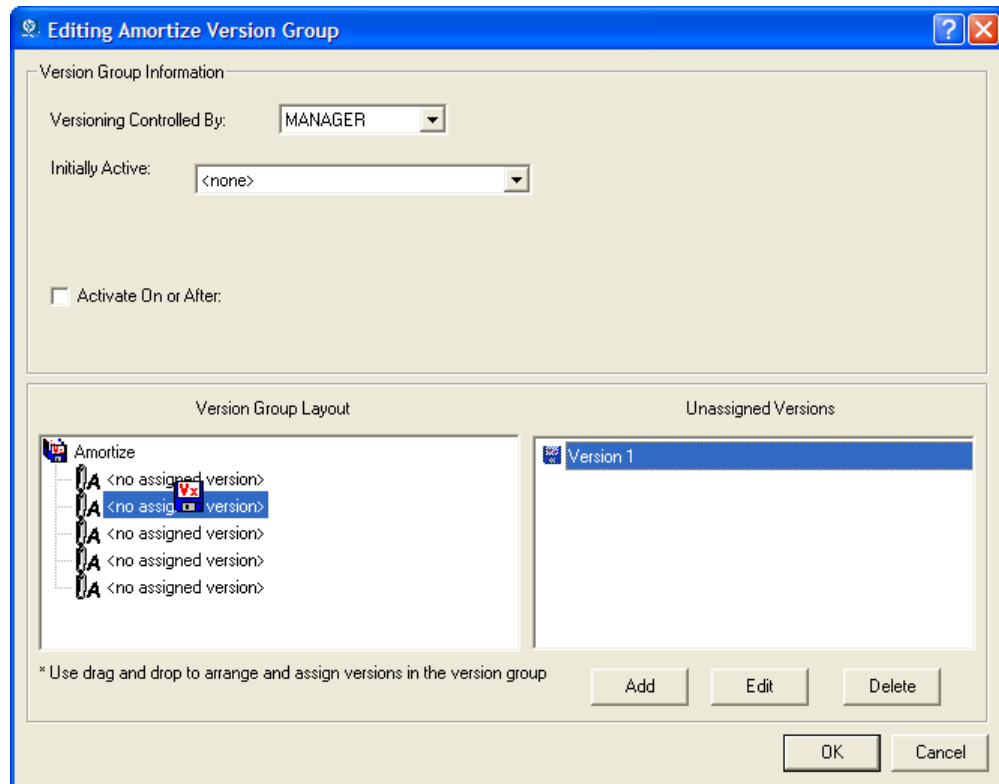
Figure 7.12 ~ Assign a Version instance to the Version Group.

Assign each of the Version instances that you created to the Version Group.

To remove a Version instance assignment

■ In the **Version Group Layout**, click a Version instance and drag it to the **Unassigned Versions** area. Then, release the mouse button.

Preparing a Version Group for Deployment

Now that you have created the Version instances and assigned them to your Version Group, you are ready to specify how you would like to deploy the versions. Use the **Version Group Information** area in the Version Group Editor to define the deployment of the versions.
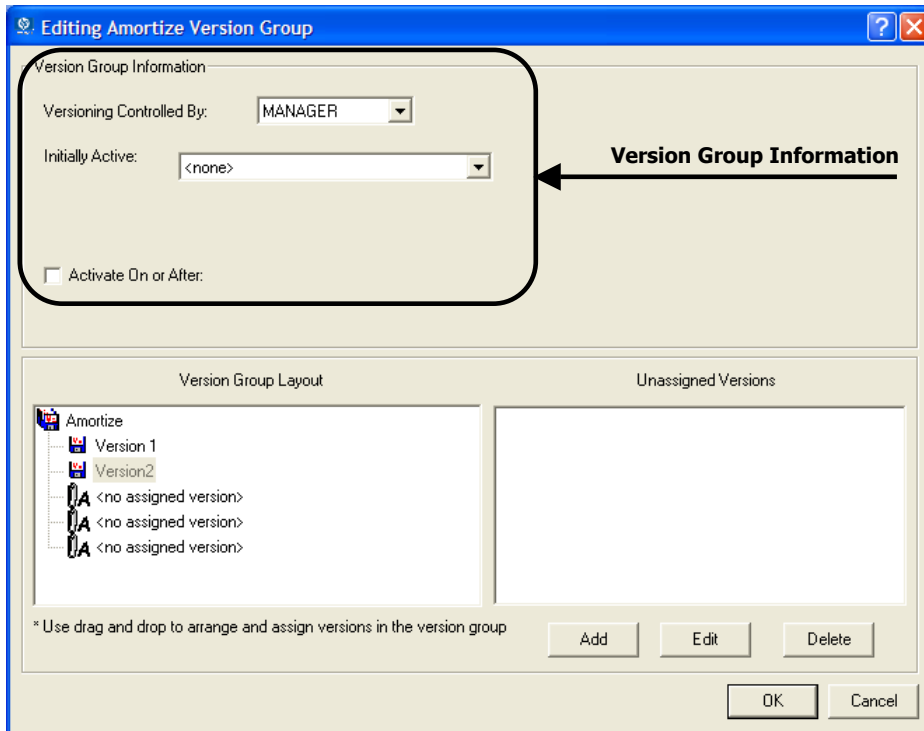
Figure 7.13 ~ Version Group Information area.

To prepare a Version Group for deployment

1. In the **Versioning Controlled By** drop-down list, select **MANAGER** or **CLIENT**.

- Select **MANAGER** if you want to control the version to be deployed.
- Select **CLIENT** if you want the subscriber to control the version to be deployed. *This is only used with the Radia Software Manager client.*

> **Note**
>
> You can *schedule* deployments of versions *only* if the Radia Configuration Server controls the versions. Therefore, to schedule deployments, you must select **MANAGER** from the **Versioning Controlled By** drop-down list.

2. In the **Initially Active** drop-down list, select the version that you want to activate on the client computer the next time the subscriber connects to the Radia Configuration Server. You can select from the versions that appear in the Version Group Layout list. The selected version is bolded in the Version Group Layout list, as shown in Figure 7.14 on page 325.

3. Select the **Activate on or After** check box to access additional controls used to delay activation of a version until a specific date and time.

> **Note**
>
> If you selected **MANAGER** in the **Versioning Controlled By** drop-down list, you *must* select the **Activate On or After** check box so that the Radia Configuration Server knows when to activate the next version.
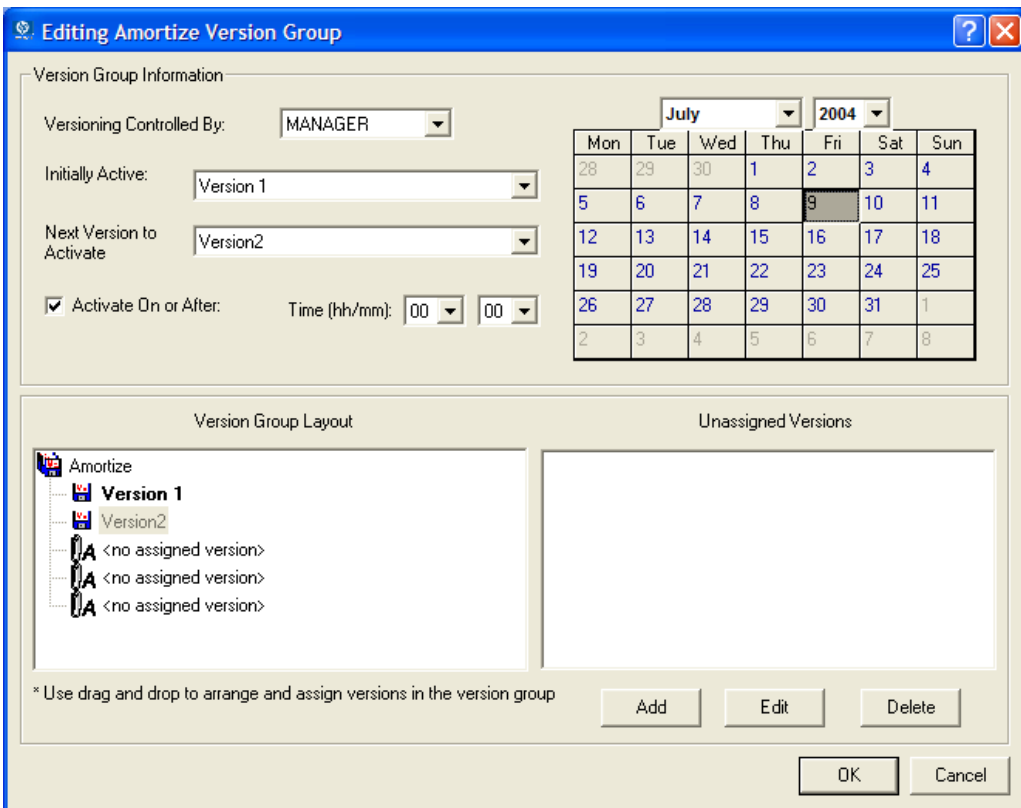
Figure 7.14 ~ The Editing Version Group dialog box.

4. In the **Next Version to Activate** drop-down list, select the version of the application that you want to activate after the initial version.

5. In the **Time (hh/mm)** drop-down list, select when you want the version to be activated.

6. Use the **Calendar** controls to set the date of deployment for the next version.

Caution

If you use the Time and Calendar controls to schedule the deployment of a version, consider the following:

- You can *schedule* deployments of versions *only* if the Radia Configuration Server controls the versions. Therefore, to schedule deployments, you must select **MANAGER** from the **Versioning Controlled By** drop-down list.

- If you selected **MANAGER** in the **Versioning Controlled By** drop-down list, you must select the **Activate On or After** check box to let the Radia Configuration Server know when to activate the next version.

If you delete a VGROUP instance, the associated timer instance will be deleted.

---

7. Click **OK** to save the information in the Version Group Editor.

8. Click **Yes** to confirm your changes.

The Version Group instance appears in the Version Groups (VGROUP) class. If you scheduled the next version to activate, Radia creates a timer instance in the Scheduling (TIMER) class and automatically connects the timer to the Version Group.

To connect the Version Group to the Service

1. Navigate to **PRIMARY.SOFTWARE.ZSERVICE**.

2. Right-click the appropriate service, in this example **Amortize**.

3. Select **Show Connections**.

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

4. Click **Version Groups (VGROUP)**.

5. Click **OK**.

The Version Group instances appear in the list view of the Radia System Explorer.
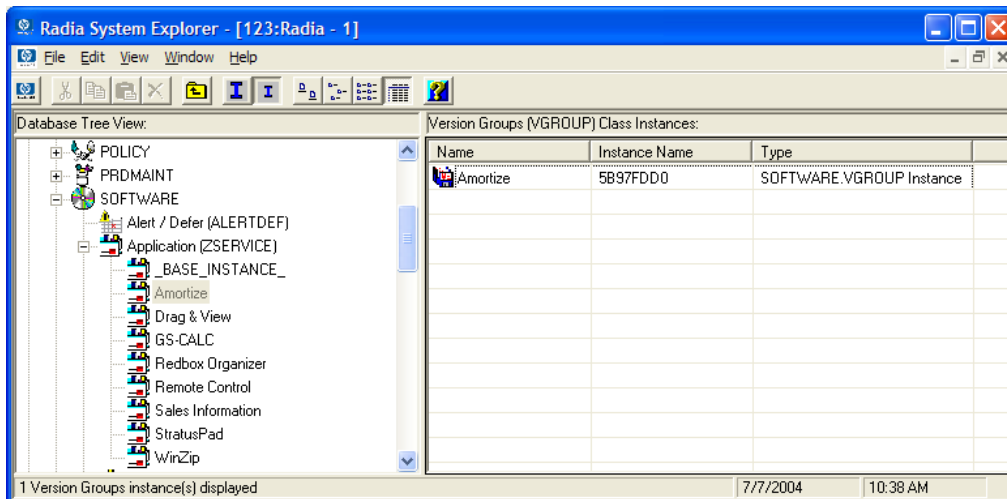
Figure 7.15 ~ Version Groups in the list view.

6. Click **Amortize** in the list view and drag it to the appropriate Application (ZSERVICE) instance (in this example, Amortize). When your cursor changes to a paper clip, release the mouse button.

The **Select Connection Attribute** dialog box opens.

7. Click **Copy**.

8. Click **Yes** to confirm that you want to connect the Amortize Version Group to the Amortize service.

9. Click **OK** when you receive a confirmation message.

> **Note**
>
> Be sure to connect the ZSERVICE to the POLICY instance for the subscribers to whom you want to deploy this.

The next time the client computer connects to the Radia Configuration Server, the initial version of the application is activated, and the compressed files for the next version will be stored on the client computer.

Editing a Version Group

After you create a version group and its instances, you may want to return to the Version Group Editor to make changes.

To edit a Version Group

1. Navigate to the Version Group instance, located in **PRIMARY.SOFTWARE.VGROUP**.

2. Right-click the appropriate Version Group instance.

3. Click **Version Group Editor**.

   The Version Group Editor opens. Modify the Version Group as necessary.

4. Click **OK** to save your changes.

   OR

   Click **Cancel** to close the Version Group Editor without saving your changes.

The Version Group (VGROUP) Class

Each instance of the Version Group (VGROUP) class defines a set of versions for an application. This class contains connections to the Versions (VERSION) class, created using the Version Group Editor in the Radia System Explorer.
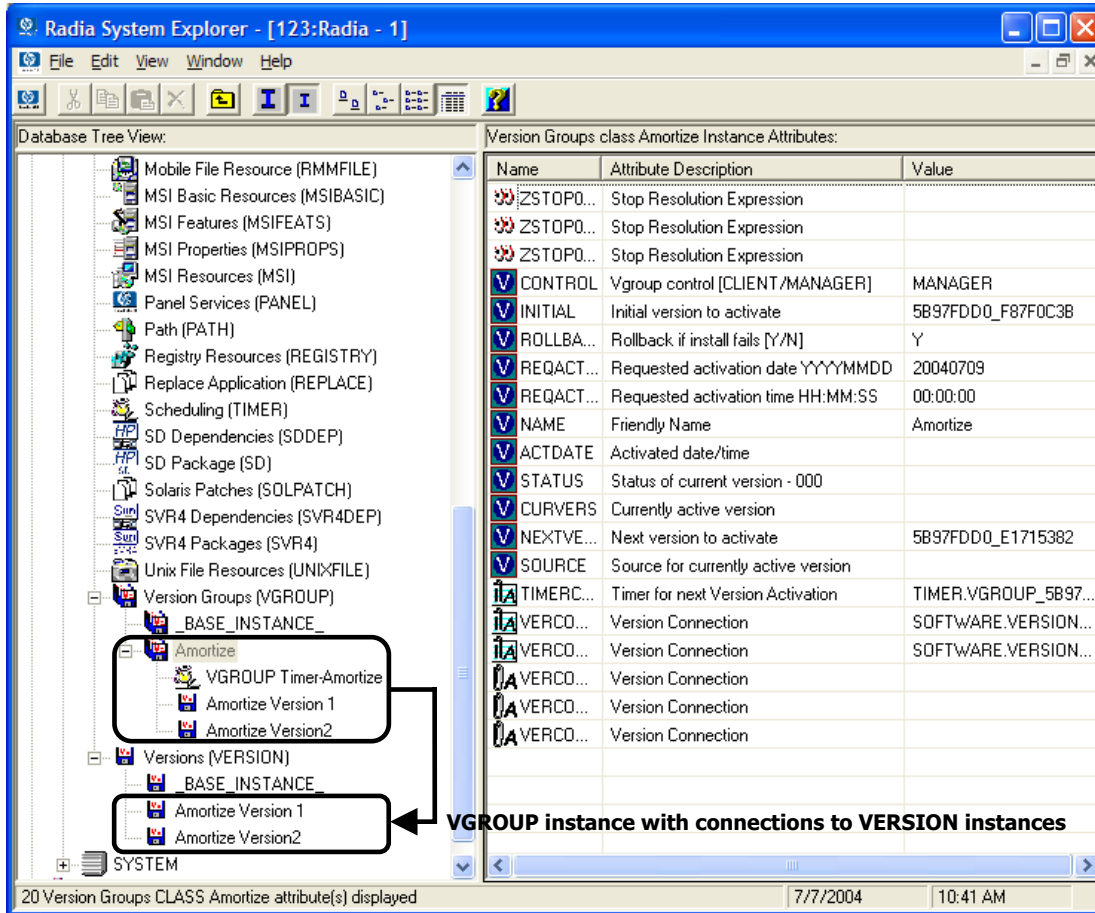
Figure 7.16 ~ Version Groups (VGROUP) class instance.

The following table describes each of the attributes in the Version Group (VGROUP) class.

Table 7.5 ~ Version Group (VGROUP) Class Attributes

Attribute	Description
ZSTOP00n	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the Radia System Explorer to set this attribute.
CONTROL	Indicates whether the Radia administrator (MANAGER) or the subscriber (CLIENT) controls which version to activate on the client computer. Use the **Versioning Controlled By** drop-down list in the Version Group Editor to set this option. **Note:** The Radia Application Manager supports Radia administrator-controlled version activation, but does not support subscriber-controlled activation.
INITIAL	Indicates which version to activate on the client computer. Use the **Initially Active** drop-down list in the Version Group Editor to set this option.
ROLLBACK	Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails. A new version may fail to deploy because of lack of sufficient disk space on the client computer, improper packaging, or failure of a method to complete successfully. By default, ROLLBACK = y.
REQACTDT	The earliest date on which a version in this version group will be activated on any client computer. If this attribute is blank, the version identified by the INITIAL attribute will be activated at the end of the Client Connect that causes the version to be transferred to the client computer. Use the calendar controls in the Version Group Editor to set REQACTDT.
REQACTTM	The earliest time, on the date specified by the REQACTDT attribute, after which a version in the version group will be activated on any client computer. The version identified by the INITIAL attribute will be activated during the next Client Connect. Use the **Time (hh/mm)** drop-down lists in the Version Group Editor to set REQACTTM.
NAME	The friendly name for the VGROUP instance. This is set when you create the instance using the Radia System Explorer.
ACTDATE	Data maintained by the Radia Client in the VGROUP object on the client computer. Do not enter a value. This is set by the Radia Client.
STATUS	Data maintained by the Radia Client in the VGROUP object on the client computer. Do not enter a value. This is set by the Radia Client.
CURVERS	Data maintained by the Radia Client in the VGROUP object on the client computer. Do not enter a value. This is set by the Radia Client.
NEXTVERS	Data maintained by the Radia Client in the VGROUP object on the client computer. Do not enter a value. This is set by the Radia Client.
SOURCE	Data maintained by the Radia Client in the VGROUP object on the client computer. Do not enter a value. This is set by the Radia Client.
TIMERCON	If you specify a "next version to activate," the Radia System Explorer automatically creates a timer and stores the connection to that timer in this attribute.
VERCON0n	Connects to each version in the version group. Each VERCON0n attribute contains a connection to one instance of the VERSION class. This is set when you assign a version to the version group in the Version Group Editor.

The Versions (Version) Class

Each instance of the version class defines one *version* of an application to be deployed and managed by Radia. Use the Version Group Editor to create Versions class instances and assign them to a Version Group.
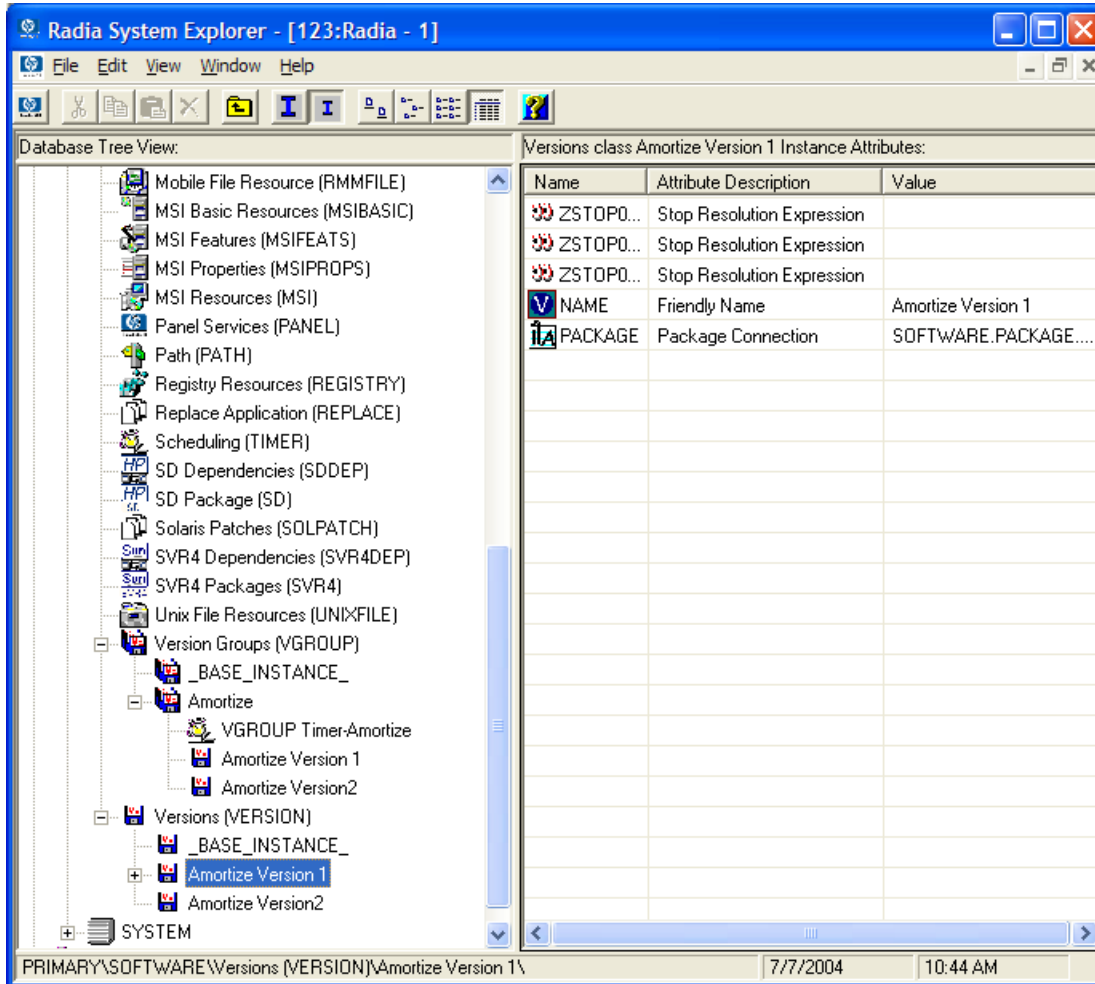
Figure 7.17 ~ Versions (VERSION) class instance.

The following table describes each of the attributes in the Versions (VERSION) class.

Table 7.6 ~ Versions (VERSION) Class Attributes	
Attribute	**Description**
ZSTOP00n	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is not skipped, and resolution continues. This is useful for assigning a version to a specific set of users. Use the Radia System Explorer to set this attribute.
NAME	The friendly name for the VERSION instance. This is set when you create the instance using the Version Group Editor.
PACKAGE	Connects to a PACKAGE class instance, which represents the packaged software for this version.

Application (ZSERVICE) Attributes

This section describes the attributes that you will see if you open an Application (ZSERVICE) instance in the Radia System Explorer. Many of the values for these attributes are set when using the Radia Administrator Workstation, such as the Radia Publisher or the New Application Wizard in the Radia System Explorer. You can also use the Radia System Explorer to modify the values of these attributes in the SOFTWARE.ZSERVICE class.

You may notice that some attributes do not have values, or their values are not displayed in the Radia System Explorer. The Radia Client uses these attributes. For example, an attribute such as INSTDATE is used to record the date the service was installed on the client computer. The value for this attribute is stored in the PROFILE file for the client computer in the Radia Database.

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
ZSTOPnnn	**Expression Resolution Method**
	Stops resolution if the expression evaluates to TRUE.
	Example: WORDPOS(EDMGETV(ZMASTER,ZOS),'WINXP WIN2K NT')=0
	This example expression will stop resolution on the instance if the client computer's operating system if the operating system is *not* Windows XP, Windows 2000, or Windows NT. In other words, the application will not be installed unless the client computer is running Windows XP, Windows 2000 or Windows NT.
ZSVCNAME	**Service Name/Description**
	Name of the service used for display in the Radia Software Manager user interface. Value is set initially in the Short Description field in the New Application Wizard.
ZSVCTTYP	**Application Target Type [A/S]**
	Indicates which Radia Client this application was packaged for, Radia Application Manager or Radia Software Manager. Value is set initially in the New Application Wizard. Possible values are **A** for Radia Application Manager and **S** for Radia Software Manager.
ZSVCMO	**Mandatory or Optional Service [M/O]**
	Designates a service as mandatory or optional. This value is set initially based on the setting for the application target type (ZSVCTTYP) in the New Application Wizard. Usually, when using the Radia Application Manager, services are marked as mandatory, **M**. When using the Radia Software Manager, services are usually marked as optional, **O**.
	If you are using Radia Application Manager *and* Radia Software Manager, you could also specify mandatory and then optional, **MO**, or optional then mandatory, **OM**. The first character indicates how the application should be handled before installation. The second character indicates how the application should be handled after installation. For example, suppose you want a Radia Software Manager subscriber to have the option of installing the application, but, once installed, want the maintenance or removal of the application to be mandatory, set ZSVCMO to be **OM**.
	Note: If you may need to edit the ZSERVICE class template, to allow you to set ZSVCMO to OM. See the Radia System Explorer Guide for more information on editing a class template. To process mandatory applications using Radia Software Manager, add "enterprisemanagement=auto" to the args.xml file.
ZSVCPRI	**Service Create Ordering [01-99]**
	Set the priority level for the service. Services are created based their priority. The lower the number the higher the service's priority. A service with ZSVCPRI set to **01** would have the highest priority while a service set to **99** would have the lowest priority.
ALWAYS	**Contains**
	Any method that you specify for this attribute is unconditionally executed when this instance is resolved.
	Example: A valid method name such as ZSYSTEM.ZMETHOD.PUTHIST_ZERROR.
ZCREATE	**Service Installation Method**
	Method that runs when the service is installed. For example, a command to start a service that was stopped to install files.
ZINIT	**Service Initialization Method**
	Method that runs when the service is initialized. For example, a command to stop a service before installing files that the service might lock.

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
ZDELETE	**Service Delete Method**
	Method to run when the service is deleted.
ZUPDATE	**Service Update Method**
	Method to run when the service is updated.
ZVERIFY	**Service Verify Method**
	Method to run when the service is verified.
ZREPAIR	**Service Repair Method**
	Method to run when the service is repaired.
PUBDATE	**Published Date of Service**
	Reserved for future use.
UPDDDATE	**Upgrade Date (Programmatic)**
	Reserved for future use.
AUTHOR	**Author Name**
	Name of the author of the service that appears in the extended information area in the Radia Software Manager user interface. Value is set initially in the Author field in the New Application Wizard.
DESCRIPT	**Application Description**
	Description of the service that appears in the properties for the service in the Service List. Value is set initially in the Long Description field in the New Application Wizard.
VENDOR	**Vendor Name**
	Name of the vendor of the service that appears in the Radia Software Manager user interface. Value is set initially in the Vendor field in the New Application Wizard.
URL	**WEB URL Name**
	Address of a Web page where the subscriber can find additional information about the service. This appears in the properties for the service in the Radia Software Manager user interface. Value is set initially in the Web URL field in the New Application Wizard.
CATGROUP	**Catalog Group Name**
	Use CATGROUP to group a set of applications into a group. You can display applications based on their group in the Radia Software Manager user interface.
PRICE	**Price**
	Type in the price of an application to be displayed to subscribers in the extended information area in the Radia Software Manager user interface.
SCHEDOK	**Update Schedule Locally [Y/N]**
	For Radia Software Manager only. Specify **Y** to allow the subscriber to change the update schedule locally. Specify **N** to maintain control on the Radia Configuration Server.
VERSION	**Version Description**
	Version of the software. This appears in the properties for the service in Radia Software Manager user interface. The value is set initially in the Version field in the New Application Wizard.

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
NAME	**Friendly Name** This name appears in the properties for the service in the Radia Software Manager user interface. The value is set initially in the Short Description field in the New Application Wizard.
OWNER	**Application Contact** *Reserved for future use.*
RUNDLG	**Dialog Processing [Y/N]** Specifies whether to enable processing of instances in the DIALOG class during the installation of the service. Specify **Y** for Yes and **N** = No. **Default**: N
REBOOT	**Install/Update/Delete/Version Chang** Used to restart the client computer based on application event. Specify your action by equating an application event to a reboot type, panel, or connect. **Event to restart on:** **AI** = Install **AD** = Deinstall **AU** = Update **AR** = Repair **AV** = Verify **Type of reboot:** **S** = Soft Boot (Default of type **Y** panel.) **H** = Hard Boot (Default of type **A** panel.) **N** = None **Type of panel:** **Q =** No panel. **A =** **OK** button only. **Y =** **OK** and **Cancel** button. **Type of connect:** **None specified:** Reboot on Machine connect (context = m). **U =** reboot on user connect only (context = u). **MU =** reboot when both machine and user parts of the service have been installed. **Example:** AI=S performs a soft boot on application installation.

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
EVENTS	**Events to Report**
	Indicates which events to report on. Specify your event by equating an application event to an event type.
	AI: Application Install
	AD: Application Deinstall
	AU: Application Update
	AR: Application Repair
	AV: Application Verify
	VA: Version Activation
	VD: Version Deactivation
	What to report on:
	S: Success
	F: Failure
	B: Both Success and Failure
	N: None
	Default: AI=B,AD=B,AU=F,AR=N,VA=F,VD=F
ERTYPE	**Event Reporting Method [O/E/X]**
	Set this attribute to send an APPEVENT object. Currently, this supports object (O) format only.
	Default: O
ADAPTIVE	**Auto Adaptability [Y/N]**
	Indicates whether the installed package is dependent on client settings that must be monitored periodically, such as plug and play devices. If the settings change, the client must reconnect to the Radia Configuration Server to get new or different components. Specify **Y** for Yes and **N** for No.
LREPAIR	**Local Repair [Y/N]**
	Enables local repair of broken applications. If an application is broken because of missing files, the files (stored locally) can be used to repair the application. Specify **Y** for Yes and **N** for No.
	Default: N
REMOVAL	**Un-Managed Behavior [A/D/U]**
	Controls how the application is managed when a service is removed.
	Set REMOVAL to **A** (Abandon) to delete the service's objects on the client, but leave the application components. The service will no longer be managed by Radia.
	Set REMOVAL to **D** (Delete) to delete the service's objects and components. The service will still be managed by Radia.
	Set REMOVAL to **U** (Unmanage) to stop management of the service by Radia. Neither the objects nor the components are deleted. This applies only to optional applicatons (ZVSCMO set to O) that are removed based on entitlement policy.
	If a subscriber removes an optional application, the service's objects are always removed no matter what REMOVAL is set to.
	Default: D

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
RECONFIG	**Reconfiguration Enabled [Y/N]** Indicates whether an application can be relocated after it has been installed. Specify **Y** for Yes and **N** for No. For example, this allows you to move an application that was installed on the C drive to the D drive without removing and re-installing the application.
ZSVCCAT	**Service Visible in Catalog [Y/N]** Specifies whether the service is visible in the Radia Software Manager Catalog. For optional applications, the default is Y. For mandatory applications, the default is N. Specify **Y** for Yes and **N** for No if you want to override these defaults.
UIOPTION	**Progress Indicator [NONE/FULL/INFO]** Controls whether the service status window appears. Possible values are: **NONE** = No interface appears. **FULL** = Interface appears and **Cancel** button is available. **INFO** = Interface appears with no option to cancel.
CACHE	**App Element Caching [Y/N]** Enables element caching. Specify **Y** for Yes and **N** for No. **Default**: N
CACHELOC	**CACHE Location On Client** *For Windows Installer applications only.* Location of the folder on the client computer that is used to cache the compressed application files needed for the product. Radia support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If **CACHELOC=C:\progra~1\Novadigm** and **PRODGUID = 12345_XXXX,** then the cache folder would be: **c:\progra~1\Novadigm\12345_XXXX\cache.** **Note**: The folder \cache is automatically appended to PRODGUID. If you are not deploying a Windows Installer-enabled application, the files will be cached in IDMDATA. **Default**: _UNDEF_
CACHELIM	**Percnt Disk Limit For Cache** *For Windows Installer applications only.* Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.
ZDISCONN	**Disconnect on Install [Y/N]** Allows the client to disconnect from the Radia Configuration Server if there is an open session with the Radia Configuration Server. Specify **Y** to disconnect the client from the Radia Configuration Server. Specify **N** to keep the client connected to the client from the Radia Configuration Server. **Default**: N

Table 7.7 ~ Modifiable SOFTWARE.ZSERVICE Attributes	
Attribute	**Description**
ZSYSACCT	**Install under System Account[Y/N]**
	Specifies whether to install the service under the system account or the user's account. Specify **Y** to install the application using the system rights. Specify **N** to install the application using the rights of the logged on user.
	Default: N
MCELIGBL	**Service Multicast Eligible[Y/N]**
	Indicates if the application is eligible for multi-casting. Specify **Y** for Yes and **N** for No.
	Default: Y
RSTRSIZE	**Download restart threshold (bytes)**
	Use the RSTRSIZE attribute in the appropriate ZSERVICE class instance to control which files are enabled for check point restart based on the amount of data being downloaded (in bytes).
ZSVCMODE	**Application Context [M/U/MU/EMU]**
	Set ZSVCMODE to **M** if the service has only machine components. This service will be ignored if context is set to u on the radskman command line.
	Set ZSVCMODE to **U** if the service has only user components. This service will be installed if context is set to u or is left blank on the radskman command line. You may want to set ZSVCMODE to u if the application consists only of user registry changes or user desktop shortcuts.
	Set ZSVCMODE to **MU** if the service has both machine and user components. The user connect will verify that the machine components have been installed before installing the user components. You will need to run two radskman connects, one with context set to m and one with context set to u.
	Set ZSVCMODE to EMU if the Client Connect is being made in the user context, but the machine side of the application has not yet been installed, this will force the machine connect. After the machine connect completes successfully, the user connect is initiated to install the user components. Use this for optional applications that the user controls through the Radia Software Manager.
	Leave ZSVCMODE blank to treat the service as single mode that can be installed independently by the machine or the user. In other words, install the entire service ignoring the component's ZCONTEXT.

Reporting Attributes in ZSERVICE

Some of the attributes in the ZSERVICE class are calculated. They are updated when the service is installed, verified, updated, repaired, or deleted and reported in the client computer's service objects. These attributes should *not* be modified using Radia System Explorer.

Table 7.8 ~ Calculated ZSERVICE Attributes – DO NOT MODIFY

Attribute	Description
ZSVCCSTA	**Service Status on Client** Status code for the service. Used to determine why files for a service may not be deployed correctly. Values range from 000-999.
SIZE	**Application Size - Uncompressed** The size of the uncompressed application displayed to the subscribers in the extended information area in the Radia Software Manager user interface. Since this is a calculated field, do not modify it. It is the cumulative value of the SIZE defined in the PACKAGE class.
COMPSIZE	**Application Size - Compressed** The size of the compressed application displayed to the subscribers in the extended information area in the Radia Software Manager user interface. Since this is a calculated field, do not modify it. This is the cumulative value of the COMPSIZE defined in the PACKAGE class.
ZAVIS	**Available, Verified, Installed, Sync F** The Radia Client manages and maintains this attribute to show the different states of the application in the catalog. The four states are: **Available** indicates whether a service is available from the Radia Configuration Server. **Verified** indicates whether a service has been verified. **Installed** indicates whether the service has been installed. **Synchronized** indicates whether the installed service has all of the latest changes from the Radia Configuration Server. The possible values for each are: **Y** = Yes **N** = No **X** = Unknown
VERDATE	**Verified Date of Service** Indicates when the application was last verified (in local time) on the client computer. The Radia Client manages and maintains this attribute. This is displayed to the subscribers in the extended information area in the Radia Software Manager user interface. This attribute is useful for reporting purposes. The value is in the format of MMM DD,YYYY HH:MM:SS. **Example**: Jul 28, 2003 16:10:00
UPGDATE	**When Application was Upgrade on De** The Radia Client manages and maintains this attribute. It indicates when the application was last updated (in local time) on the client computer. This attribute is useful for reporting purposes. The value is in the format of MMM DD,YYYY HH:MM:SS. **Example**: Jul 28, 2003 16:10:00
INSTDATE	**Installed Date** Indicates when the application was installed (in local time) on the client computer. The Radia Client manages and maintains this attribute. This is displayed to the subscribers in the extended information area in the Radia Software Manager user interface. This attribute is useful for reporting purposes. The value is in the format of MMM DD,YYYY HH:MM:SS. **Example**: Jul 28, 2003 16:10:00

Table 7.8 ~ Calculated ZSERVICE Attributes – DO NOT MODIFY	
Attribute	**Description**
DELDATE	**Delete Date** Indicates when the application was removed (in local time) from the client computer. The Radia Client manages and maintains this attribute. This attribute is useful for reporting purposes. The value is in the format of MMM DD,YYYY HH:MM:SS. **Example**: Jul 28, 2003 16:10:00

Summary

■   Set the appropriate context, M, U, MU, or EMU, for a service using the ZSVCMODE attribute.

■   If a service requires a reboot of the client computer, use the REBOOT attribute in the Application (ZSERVICE) class.

■   Be aware of all of your service options in the Application (ZSERVICE) class.

8

Deploying Services

At the end of this chapter, you will:

■ Understand the deployment methods available in Radia.

■ Be able to use a timer to deploy a service at a pre-determined time.

■ Know how to use the Notify function to update an application, remove an application, or send an e-mail message to a subscriber.

■ Be able to deploy versioned applications.

■ Understand how Radia supports Windows Installer Applications.

This guide covers the *suggested* implementation for the Radia Application Manager. Although you will tailor this strategy to meet your organization's needs, it is recommended that you review this guide for a comprehensive understanding of the Radia Application Manager. This chapter covers deploying services.

Package Applications	Implement Entitlement Policy	Configure Operations	Prepare Services	Deploy Services	Examine Results
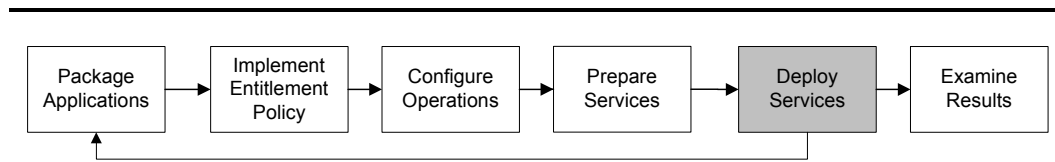
Figure 8.1 ~ Tasks completed in this guide.

About Deployment Methods

After creating a service using the Radia System Explorer, and deciding which users or groups will receive the application, you are now ready to deploy the service to your subscribers.

> **Note**
>
> If you have used other systems management software, you may be familiar with the term "job". A job is used to distribute a package. It includes a set of instructions to perform, a package containing the files or software, the targets for the job, and the schedule for carrying out the job.
>
> In Radia, you do not need to use a job. You can perform each of the steps – creating the package, defining targets (assigning users), and selecting a deployment method – individually. This provides flexibility because you can use multiple deployment methods to distribute a single package, based on the needs of your enterprise.

The following deployment methods are available in Radia.

■ **Scheduler**
Installs the service at a specific time or sets any command line to run at an interval.

■ **Notify**
Forces one or more client computers to connect to the Radia Configuration Server to install, update, or remove an application, or sends an e-mail to the subscribers of a particular service.

> **Note**
>
> The term *computer* is used to refer to a workstation or server.

Before selecting a deployment method, consider the following.

❑   Does the application need to be deployed at a certain time? If so, use Scheduler.

❑   Do you want to notify the users via e-mail when you are deploying the application? If so, use Notify.

❑   Do you want to install a new application, an update to an application, or remove an application? If so, use Notify.

❑   Are there multiple versions of the application? If so, use Version Groups. See *Chapter 7: Preparing Services* starting on page 299.

Testing Deployments

To ensure successful deployments, rigorously test your implementation.

■   Publish and deploy software in a test environment before making the software available for live deployment.

■   Test deployments to all target operating systems.

■   Test all major capabilities of the deployment, including updates to the application, removing the application from the subscriber's desktop, customized installations, and variations in hardware configurations that might affect deployment, such as shortage of disk space, physical memory, and similar constraints.

Connection Parameters (Radskman)

No matter which deployment method you choose, you will need to create a radskman command line. Some deployment methods will create the command line for you. However, you should be aware of your options. Use **radskman** to:

■   Check the status of all existing mandatory applications.

■   Add new mandatory applications.

■   Remove any mandatory applications that are no longer assigned to the subscriber.

You can specify your **radskman** command line from a command prompt, Scheduler (TIMER) instance, or Notify command. Before using any of these methods in a production environment, you should test the command line parameters you choose. The parameters can be divided into five categories:

■   Core

■   Operations

■   Machine/User

- Client Operations Profiles
- Process

In the tables below, the possible parameters for radskman are described. RAM stands for Radia Application Manager and RSM stands for Radia Software Manager. After the tables, there are examples of radskman lines for common situations.

Core

Core parameters are used in most radskman lines. These parameters include the location of your Radia Configuration Server, and how to identify the client computer for policy.

Table 8.1 ~ Radskman Core Parameters	
Parameter	**Explanation**
cat	Set **cat = prompt** to run self-maintenance, display the logon panel, and check the status of other services.
	Set **cat = y** to simply check the status of services.
	Set **cat = m** (4.0 feature) to use the local machine catalog for resolving the user's service list. This is used with context = u. Usually, this is also used with local = y.
	RAM default: prompt.
	RSM default: Depends on request type.
dname	The Radia Database domain name for the services. This is the directory under which the service catalog (ASERVICE.EDM) is stored. For example, dname=SOFTWARE.
	Software
	RAM default: SOFTWARE. If preload=y, then the default is RADSTAGE.
	RSM default: SOFTWARE. If preload=y, then the default is RADSTAGE.
IP	IP address of the Radia Configuration Server. For example, IP = 10.10.1.001
	Note: If you do not specify the IP address, Radia uses the IP address specified in the ZMASTER object stored in IDMLIB (by default, *SystemDrive***:\Program Files\Novadigm\LIB**).
	RAM default: NOVARCS (only defaults if *no* arguments are passed).
	RSM default: NOVARCS (only defaults if *no* arguments are passed).
mname	Name of the Radia Configuration Server. For example, mname=RADSVR01.
	RAM default: Radia (defaults to RADSTAGE for preload).
	RSM default: Radia (defaults to RADSTAGE for preload).
port	Radia Configuration Server port. The default for this is 3464.
	Note: If you do not specify the port, Radia uses the port specified in the ZMASTER object stored in IDMLIB (by default, *SystemDrive*:\Program Files\Novadigm\LIB\).
	RAM default: 3464.
	RSM default: 3464.
sname	Specifies the service that you want to process. If you do not specify a service, then all mandatory services are processed.

Table 8.1 ~ Radskman Core Parameters

Parameter	Explanation
startdir	Specifies the IDMLIB starting directory. **Note:** If uid is set on the command line, and startdir is not, then the startdir will be set to the same value as uid. If you specify a UID on the command line, we recommend specifying the STARTDIR as well. • Set **startdir = $MACHINE** to use the computer name. • Set **startdir = $USER** to use the currently logged on subscriber. • Set **startdir =** *value* to specify a custom starting directory. If *value* contains embedded spaces, enclose the entire name in double quotes. **RAM default**: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). **RSM default**: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). Radia Software Manager does not pass a context by default.
uid	Identification used to identify the current session. **Note:** If uid is set on the command line, and startdir is not, then the startdir will be set to the same value as uid. If you specify a uid on the command line, we recommend specifying the startdir as well. • **uid = $MACHINE** identifies the current session by the name of the computer. • **uid = $USER** identifies the current session by the name of the user currently logged on. • **uid=***custom* is used to identify the current session by a custom value that you specify. **RAM default**: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). **RSM default**: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). If you do not specify a context, the user ID, Radia uses the LOCALUID specified in the ZMASTER object stored in IDMLIB (by default, ***SystemDrive*:\Program Files\Novadigm\LIB**).

Operations

These parameters influence how client will connect. Features include computer restart handling, log specifications, and the display options for the subscriber.

Table 8.2 ~ Radskman Operations Parameters

Parameter	Explanation
ask	Set **ask = y** to prompt the subscriber before restarting the computer. This allows subscribers to save their work and close applications before the computer restarts. Set **ask = n** to restart the computer without prompting the subscriber. This is useful for unattended computers. **RAM default**: Y if Radia System Tray is running. N if Radia System Tray is *not* running or there are no users logged on. **RSM default**: Y

Table 8.2 ~ Radskman Operations Parameters

Parameter	Explanation
hreboot	Set **hreboot = y** to allow **radskman** to handle a computer restart if it is required by the service. Set to **p** to power off the computer. If set to **p**, the client computer will shut down no matter what the reboot settings are for a particular service. **Note**: This replaces handle_reboot. **RAM default**: Y **RSM default**: N
ind	Set **ind=n** to hide the status indicator for each service. Set **ind=y** to show the status indicator for each service. **RAM default**: Y **RSM default**: Y
jobid	Use jobid to further describe the source of this command line. It shows up in the APPEVENT, IDENTITY, PREFACE, SYNOPSIS as JOBID. **RAM default**: UserConnect if started in a USER context. MachineConnect if started in SYSTEM context. **RSM default**: UserConnect if started in a USER context. MachineConnect if started in SYSTEM context.
log	Specifies the name of the log stored in the IDMLOG directory.
logsize	Specifies the size of the log file in bytes. When the logsize is reached, a backup file (.BAK) is created. By default, this file is connect.bak. If a backup file already exists, it will be overwritten. **RAM default**: 1000000 bytes **RSM default**: 1000000 bytes
rtimeout	Specify number of seconds to wait if a reboot panel has been requested for a service before rebooting the client computer. This will allow a subscriber time to save and close applications before a reboot.

Machine/User

Use these parameters when using applications with machine and user components, or when you have multiple users on the same client computer. These parameters can control frequency of connection to the Radia Configuration Server, display of the user logon panel, and when to send objects to the Radia Configuration Server.

Table 8.3 ~ Radskman Machine/User Parameters

Parameter	Explanation
cat	Set **cat = prompt** to display the logon panel, and check the status of other services. Set **cat = y** to simply check the status of services. Set **cat = m** (4.0 feature) to use the local machine catalog for resolving the user's service list. This is used with context = u. Usually, this is also used with local = y. **RAM default**: prompt **RSM default**: Depends on request type
context	Set **context = m** when installing an application in the machine context. If context = m then the following defaults are assumed: uid=$machine startdir=system cat=prompt ulogon=n Set **context = u** when installing an application in the user context. If context = u then the following defaults are assumed: startdir=$user uid=$user cat=prompt ulogon=y **RAM default**: If started with a user logged on, the context defaults to u. If no user is logged on, then context defaults to m. **RSM default**: There is no default for Radia Software Manager. All components are processed.
flushu	If you are using **local=y**, set **flushu=y** on user connects (context=u) to send reporting objects up to the Radia Configuration Server at the end of the local connect for immediate feedback. This is the default behavior on user connects. If you are using **local=y**, set **flushu = n** on a user connect (context=u) if you do not want the objects sent to the Radia Configuration Server. Be aware that the user's objects will continue to grow until they are sent to the Radia Configuration Server. On a machine connect (context=m), set **flushu=a**, if you want to send all user's reporting objects to the Radia Configuration Server. **RAM default**: Y **RSM default**: Y
local	Set this to **y** to install resources for the user's services from the local client computer. Use this only with **context = u**. Usually, this is used with **cat = m**.
machfreq	(4.0 Feature) Use this variable to prevent Radia from running every time a client computer reboots. Set this to a positive integer, n, to run a machine connect only if it has been n hours since the last time a machine connect ran. This value ensures that the Radia Client will not run more than once within the specified timeframe to reduce the number of ROM commits on a thin client computer. If you set MACHFREQ to 0, the machine connect will run on *every* reboot of a thin client.
ulogon	*Only used if cat = prompt.* Set **ulogon = n** to hide the logon panel. **Note**: If using Radia System Tray, set **ulogon = n**. This will display the Radia logon panel which is not supported by Radia System Tray. **RAM default**: N **RSM default**: Y

Table 8.3 ~ Radskman Machine/User Parameters	
Parameter	**Explanation**
userfreq	*Only used if* **context=u***.*
	Use this variable to prevent Radia from running every time a user logs into the client computer.
	Set this to **0** to run a user connect only if a machine connect has run since the last user connect.
	Set this to a positive integer, *n*, to run a user connect if a machine connect has run *or* it has been *n* hours since the last time a user connect ran.
	If the value of userfreq is blank or not supplied, then a user connect will run every time a Client Connect is run with context = u.

Client Operations Profiles

These parameters are used for specifying how to use Radia Client Operations Profiles. For more information, see the chapter, *Radia Client Operations Profiles*, in this guide.

Table 8.4 ~ Radskman Client Operations Profiles Parameters	
Parameter	**Explanation**
cop	(4.0 Feature) Set to Y to enable Client Operations Profile resolution for this Client Connect only. Set to N to disable Client Operations Profiles resolution for this Client Connect only. If the RADSEUP object exists the methods will not be run, but the other settings (from the CLIENT.SETTINGS class) will be used. Set to M to run a "method" connect. In other words, use the attributes specified in the RADSETUP object, including EXBSETUP, EXASETUP, EXBEXIT, EXBOUTBX, and CMETHOD, but do not do Client Operations Profiles resolution.
	RAM default: N
	RSM default: N
datauri	(4.0 Feature) If you want to override the use of the SAP object for the Data Type, add datauri to the **radskman** command line. Datauri should be in the same format as the Universal Resource Identifier. For the syntax of this parameter, see *Table 6.2 ~ URI Examples* on page 270.
product	(4.0 Feature) If you used the SAP.PRODUCT attribute to identify that a SAP can only be used with a specific product, specify that product using this parameter. For example, if SAP.PRODUCT is set to RIM, set product=RIM on the radskman command line. Specify multiple product filters separated by a comma.
rcsuri	(4.0 Feature) If you want to override the use of the SAP object for the RCS Type, add rcsuri to the **radskman** command line. Rcsuri should be in the same format as the Universal Resource Identifier. For the syntax of this parameter, see **Table 6.2 ~ URI Examples** on page 270.

Process

Process parameters involve service processing such as whether to repair or add applications on the current Client Connect. These parameters also allow you to specify criteria for service processing, sending application data to a Radia Proxy Server, and handling SSL security for your Radia Clients.

Table 8.5 ~ Radskman Process Parameters	
Parameter	**Explanation**
Add	Set this to **N** if you do not want to install applications during this Client Connect. **RAM default**: Y **RSM default**: Y
autofix	Set **autofix = y** to automatically repair any broken applications. Set **autofix = n** to prevent broken applications from being fixed. **RAM default**: Y **RSM default**: Y
catexp	(4.0 Feature) Use this parameter to process applications based on a particular attribute in the ZSERVICE class. Use the format *<attribute name>*:*<value>*. Specify multiple OR conditions with a forward slash (/). For example, to process only applications that have the ZSERVICE.CATGROUP attribute set to finance, set catexp=catgroup:finance.
del	Set this to **N** if you do not want to delete applications during this Client Connect. **RAM default**: Y **RSM default**: Y
merge	Set merge equal to an object name to have all variables in that object includes in the ZMASTER object. Do this to send the variables to the Radia Configuration Server.
mnt	(4.0 Feature) Set to Y to process Radia Client Self Maintenance on this connect. **Note**: Client Self Maintenance will not be deployed unless you set mnt to Y. **RAM default**: N **RSM default**: N
preload	Use this for staging server preload. Specify the location of directory to copy the files to. If you do not need or want to specify a different data directory, setting **preload=y** uses the IDMDATA directory specified in NVD.INI.
rep	Set this to **N** if you do not want to repair applications during this Client Connect session. **RAM default**: Y **RSM default**: Y
sendcat	Set this to **y** to send the service list, stored in the client computer's ASERVICE object, to the Radia Configuration Server at the end of the Client Connect so that additional analysis can be done on the service list.
sslmgr	Specifies the hostname or IP address of the Radia Configuration Server. **Note:** To perform client self-maintenance over a secure channel (SSL), add the flag, **::sm**, to the end of the SSL Manager IP address.
sslport	Specifies the port for SSL communications (normally, 443).

Table 8.5 ~ Radskman Process Parameters	
Parameter	**Explanation**
upd	Set this to **N** if you do not want to update applications during this Client Connect session. **RAM default**: Y **RSM default**: Y
ver	Set this to **N** if you do not want to verify applications during this Client Connect session. **RAM default**: Y **RSM default**: Y

radskman Examples:

The following examples are provided to illustrate common uses of **radskman**.

`radskman ip=10.10.10.15,port=3464,mname=radia,dname=software,cat=prompt`

Performs a first catalog refresh which brings down the catalog (aservice.edm), runs self-maintenance, does not display the user logon panel if using Radia Application Manager, and processes all mandatory applications:

`radskman ip=test.corp.com,port=3464,mname=radia,dname=software,cat=prompt,uid`
`=$machine,ulogon=n,ind=n`

Performs a full connect for user <machine name> silently with no user logon panel or progress indicator panels. This is a typical command used by a daily timer. Note: the ip= parameter can be a DNS name or IP address.

`radskman ip=10.10.10.15,port=3464,mname=radia,dname=software,cat=n,autofix=n`

Verify mandatory applications **without** updating the catalog, running self-maintenance, or repairing broken applications. Note: This machine must perform a first refresh catalog using cat=prompt at least once to bring down the catalog prior to using cat=n:

`radskman ip=10.10.10.15,port=5004,mname=radia,dname=software,cat=y,sname`
`=WINZIP`

Install a single application with the service name of WINZIP, while only updating the catalog. Note: In this example, the Radia Configuration Server uses a custom port number.

`radskman ip=10.10.10.15,port=3464,mname=radia,dname=software,cat=prompt,`
`hreboot=Y,ask=Y`

Process all mandatory applications, handle reboot requests, and prompt the user with a panel to confirm the reboot request:

`radskman ip=10.10.10.15,port=3464,uid=STAGER,preload=Y`

Silently preload a Radia Staging Server using the default location of RADSTAGE. This syntax is often run by a daily or weekly timer instance that is deployed to a Radia Client that is co-located on the same machine as the Radia Staging Server. Note: The preload parameter automatically suppresses the user logon panel and progress indicator panels. The preload parameter also does not run any methods (ZCREATE, ZVERIFY, etc.) or evaluate any expressions on the Radia Configuration Server.

```
radskman ip=10.10.10.15,port=3464,uid=STAGER,preload=d:\stager,ind=Y
```

Preload a stager using a location of d:\stager and display the progress indicator panels. Note: If the Radia System Tray feature is enabled, then the progress indicator will be displayed in the Radia System Tray information bubble. If the Radia System Tray is disabled, then the progress indicator will be displayed in a separate panel.

```
radskman context=m
```

Perform a machine connect. Because context was specified as m, and no other parameters were passed the following default values are used: ip=NOVARCS, port=3464, uid=$machine, startdir=system, cat=prompt, ulogon=n, mname=Radia, dname=software.

```
radskman context=u
```

Perform a user connect. Because context was specified as u, and no other parameters were passed the following default values are used: ip=NOVARCS, port=3464, uid=$user, startdir=$user, cat=prompt, ulogon=y, mname=Radia, dname=software.

```
radskman context=u,userfreq=12
```

Perform a user connect only if a machine connect has occurred since the last user connect and there has been at least 12 hours since the last user connect.

Deployment Methods

The following section covers each of the deployment methods in detail. Select the appropriate method for your subscribers. Remember, you can use multiple deployment methods to distribute a single application.

Scheduling Deployments

The Radia Scheduler service allows you to deploy a service at a specific time. The Radia Scheduler service, **radsched**, is installed with the Radia Application Manager on the client computer.

On Windows NT and Windows 2000 systems, the Radia Scheduler runs as a system service that starts automatically. On Windows 95 and Windows 98 systems, a shortcut to **radsched** is installed in the Startup group.

> ### Note
>
> To stop the Radia Scheduler on Windows 95 or Windows 98, run MSCONFIG and select the **Startup** tab, or press CTRL+ALT+DEL to end the task. You will not see the Radia Scheduler in the Startup Folder in Program Files.

To schedule deployment of a service, configure a timer in the Radia Database on the Radia Configuration Server. When the client computer connects to the Radia Configuration Server, the timer information is transferred from the Radia Configuration Server to the client computer. The information is transferred in an object, called ZTIMEQ.
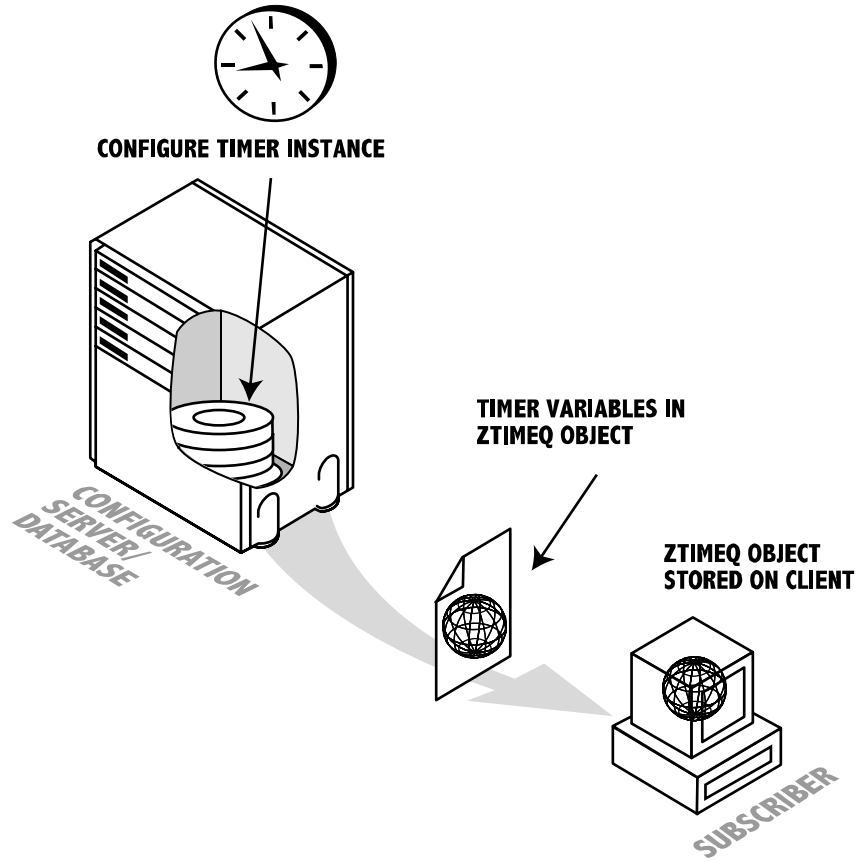
Figure 8.2 ~ Transferring the timer instance.

This section covers how to create and configure a timer, and then connect it to the service that you want to deploy. However, before creating and configuring a timer, consider the following.

❑ What time of day should the timer expire? Be sure to consider network traffic.

❑ How often do you want the timer to expire? Do you want the timer to expire daily, weekly, hourly, etc.?

❑ Does the timer need to expire more than once? For example, do you need to install the application only one time? Or, are you creating a timer that will check for mandatory applications every so often?

❑ What should happen when the timer expires? For example, do you want to launch, install, remove, or update an application?

Scheduled Deployment Strategy

One of the *suggested* strategies for implementing the Radia Application Manager client is the scheduled deployment strategy. This strategy installs an initial set of mandatory applications when you install the Radia Application Manager, and transfers a timer to the client computer that checks for new mandatory applications at the specified interval.

In this section, we will create a sample timer that updates all mandatory services on a weekly basis. In order to alleviate network congestion, the timer will expire randomly between 5:00 PM and 7:00 PM. Use the information in this section to configure timers based on your needs.

Creating a Timer

To create a timer, use the Radia System Explorer to create a Scheduling (TIMER) instance in the SOFTWARE domain.

To create a new timer in the SOFTWARE domain

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

3. The **Radia System Explorer** window opens.

4. Double-click **PRIMARY**.

5. Double-click **SOFTWARE**.

6. Right-click **Scheduling (TIMER)**.
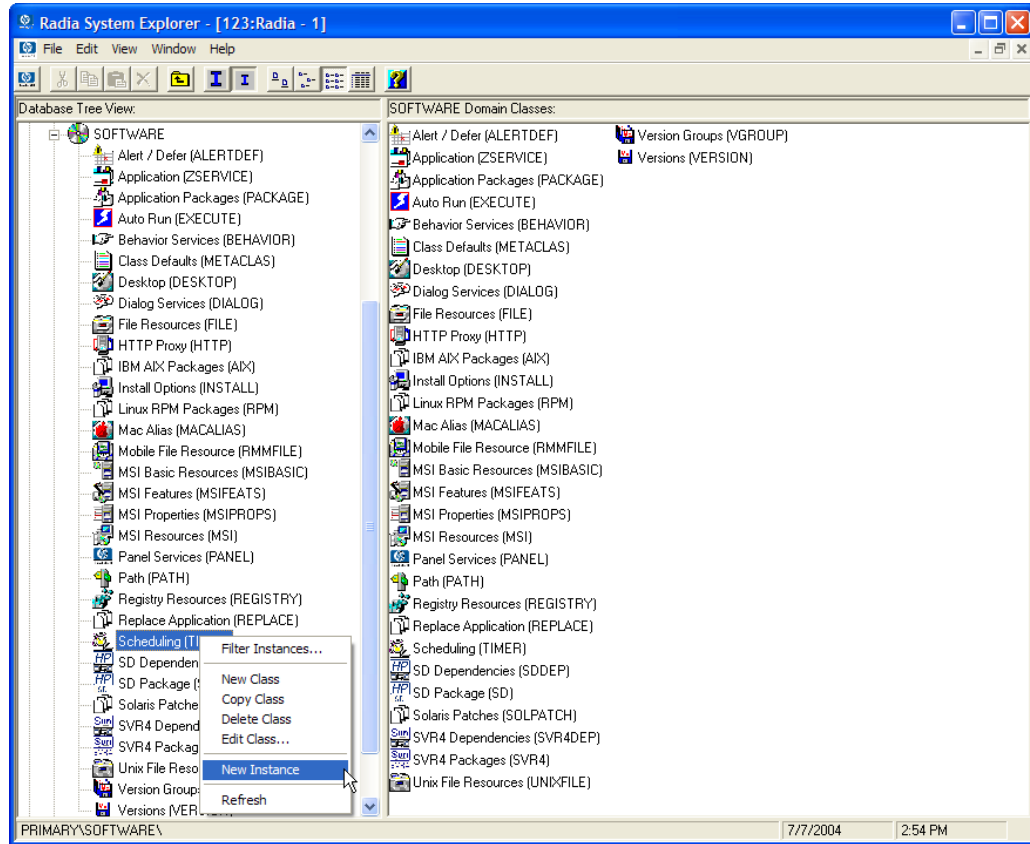
Figure 8.3 ~ Timer shortcut menu.

7. Select New Instance.

8. The **Create Instance** dialog box opens.
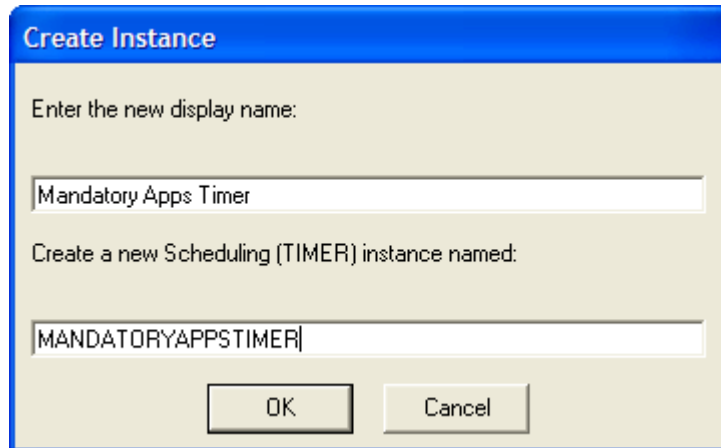
Figure 8.4 ~ Create Instance dialog box.

9. Type a name for the new timer instance, such as **Mandatory Apps Timer**.

10. Click **OK**.

The Timer instance appears in the **Scheduling (TIMER)** class.
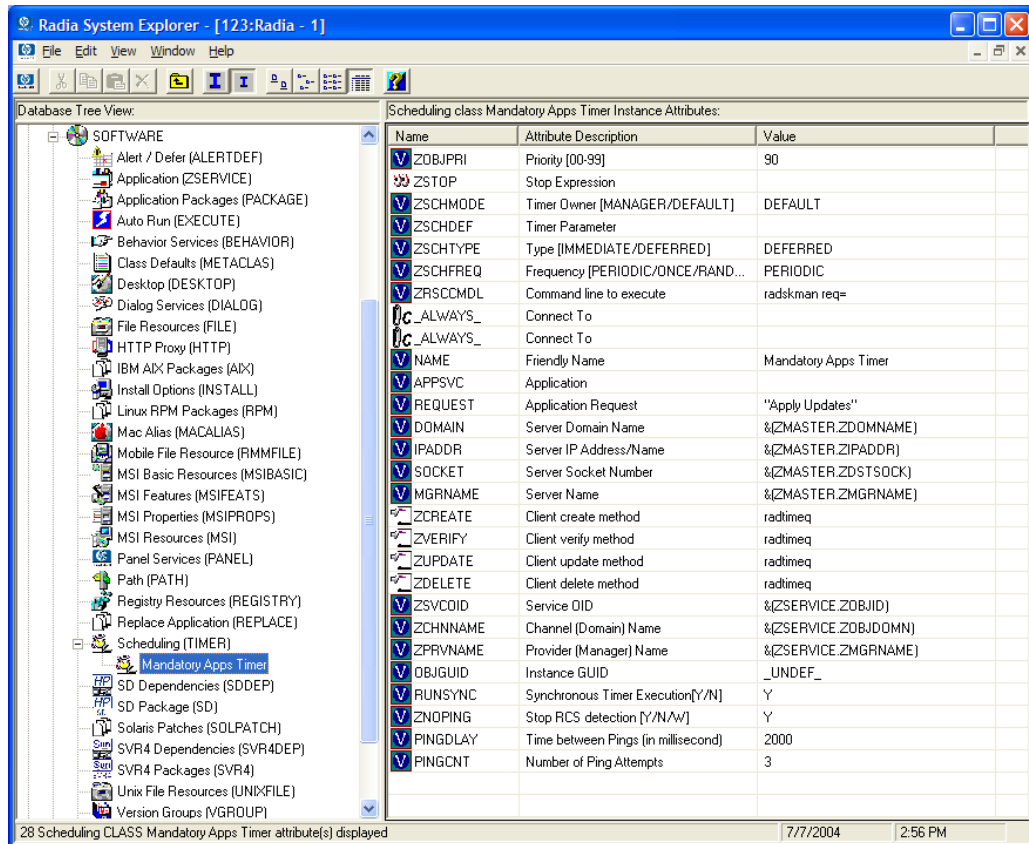
Figure 8.5 ~ Mandatory Apps Timer is created.

The Scheduling (TIMER) Class Attributes

The attributes in the Timer instance contain the information needed to execute the timer on the client computer. The following table describes these attributes.

Table 8.6 ~ Scheduling (TIMER) Attributes

Attribute	Usage
ALWAYS	Stores connections to other instances.
NAME	The friendly name for this instance.

Table 8.6 ~ Scheduling (TIMER) Attributes

Attribute	Usage
NETAVAIL	If set to Y, check for network availability before executing the Timer instance. If the network is not available, network availability will be checked every time the timer wakes up until the network is available.
	If set to N, the Timer instance will be executed without checking for network availability.
	If set to W, check for network availability before executing the Timer instance. If the network is not available and the time window's end limit has been reached, the timer will wait until the next time window before checking for network availability again.
	The default is N.
PINGDLAY	If ZNOPING is set to N, PINGDLAY specifies the time in milliseconds between pings. The default is 2000.
PINGCNT	If ZNOPING is set to N, PINGCNT specifies number of ping attempts. The default is 3 attempts.
RETRYFLG	Set to Y to retry the command up to the number of times specified in RETRYLMT, ignoring the end time for the timer. Set to W to retry the command up to the number of times specified in RETRYLMT, but stop retrying after the specified limit time has passed. Set to N to not retry. Note: a return code other than 200 will indicate success, and stop the retries.
RETRYINT	Specify number of minutes to wait between command executions. RETRYFLG must not be set to N.
RETRYLMT	Specify the number of times it to retry the command. Set this to 0 will retry until the command succeeds. RETRYFLG must not be set to N.
RETRYRC	Specify return codes that qualify for the retry logic. If this variable does not exist or is blank, RETRYRC will default to 200. A return code of 200 means that there was a fatal error due to a network connection failure with the Radia Configuration Server. If you populate this attribute, and a return code of 200 qualifies for a retry, be sure to specify 200 in the list.
	Example: RETRY = 200, 202, 209

Table 8.6 ~ Scheduling (TIMER) Attributes

Attribute	Usage
ZNOPING	Use this attribute to control automatic sensing of a network connection between the client computer and the Radia Configuration Server. The default is Y.
	An expired timer continually evaluates whether communications with the Radia Configuration Server can be established. When communications are established, the command line associated with the timer is executed. After executing the command line, the Scheduler service resumes normal evaluation of whether the timer has expired again.
	• If the ZNOPING attribute *does not exist* in the ZTIMEQ object, the Scheduler service does *not* ping the Radia Configuration Server.
	• Set ZNOPING to **Y** if you want to prevent the Scheduler service from pinging the Radia Configuration Server. This is especially useful for mobile users.
	• Set ZNOPING to **N** if you want the Scheduler service to ping the Radia Configuration Server.
	• If the Radia Configuration Server is pinged successfully, the command in ZRSCCMDL executes and the ZPENDING attribute, in the client's ZTIMEQ object, is set to N, to indicate that the Scheduler service does not need to ping the Radia Configuration Server again.
	• If the Radia Configuration Server is *not* pinged successfully, the timer is not processed any further, and the ZPENDING attribute value remains **Y**, to indicate that the next time the Scheduler service "pops", it should ping the Radia Configuration Server again.
	• Set ZNOPING to **W** if you are specifying an end limit in the ZCHDEF attribute. The Scheduler will ping the Radia Configuration Server before executing the command. If the Radia Configuration Server is unavailable, then the ZPENDING flag will be set to "W". If the ZSCHEDEF has a limit time, then when that time passes, the ZPENDING flag will be set to N, and the Scheduler will not attempt to execute the command until its next scheduled time.
ZRSCCMDL	Use this attribute to specify the command line that is executed on the client computer when the timer expires.
	Use radskman to verify and update Radia-managed mandatory applications. See Connection Parameters (Radskman) for a complete list of the parameters and examples.
ZSCHDEF	Use this attribute to specify when the timer expires. The syntax for this attribute varies depending on the frequency, which can be DAILY, HOURLY, INTERVAL, MONTHLY, MONTHDAY, NUMDAY, STARTUP, WEEKDAY, WEEKLY.
	See *Specifying When the Timer Expires* on page 362 for instructions on how to set ZSCHDEF.
ZSCHFREQ	Use this attribute to specify how often the timer should expire.
	• Set ZSCHFREQ to ONCE if you want the timer to expire one time.
	• Set ZSCHFREQ to PERIODIC if you want the timer to expire repeatedly.
	• Set ZSCHFREQ to RANDOM if you want the timer to expire in random intervals.
	See *Deploying Applications over a Period of Time* on page 364 for more information.

Table 8.6 ~ Scheduling (TIMER) Attributes

Attribute	Usage
ZSCHTYPE	*Used only when ZSCHFREQ = PERIODIC.* Set ZSCHTYPE to DEFERRED to indicate that the first time an event is attempted to be launched, it will be deferred until the *next* scheduled time, no matter when the timer instance is evaluated. This was designed to handle the case of a daily 4am (non-peak) scheduled event that is sent to the client computer during the day. If it was not deferred, it would launch during the day instead of "waiting" until the next morning. **Example 1:** Suppose you create and deploy a timer with the ZSCHDEF = DAILY(&ZSYSDATE,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: • Before 4:00:00, the command in the instance will be executed the same day at 4:00:00. • After 4:00:00, the command in the instance will be executed immediately. If ZSCHTYPE = DEFERRED and it is: • Before 4:00:00, the command in the instance will be executed the *next* day at 4:00:00. • After 4:00:00, the command in the instance will be executed the *next* day at 4:00:00. **Example 2:** Suppose you create and deploy a timer with the ZSCHDEF = WEEKDAY(FRIDAY,4:00:00) If ZSCHTYPE = IMMEDIATE and it is: • Not Friday or Friday and before 4:00:00, the command in the instance will be executed on Friday at 4:00:00. • Friday and after 4:00:00, the command in the instance will be executed immediately. If ZSCHTYPE = DEFERRED and it is: • Not Friday or Friday and before 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00. • Friday and after 4:00:00, the command in the instance will be executed a week later on Friday at 4:00:00.
ZSTOP	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is accepted, and resolution continues. This is useful if you want to set conditions on which of your subscribers receive the timer.

The values for the following attributes are set from the BASE INSTANCE of the TIMER class and *should not be edited*.

Attribute	Usage
RUNSYNC	Specifies if synchronous timer execution will take place. The default value is Y.
ZOBJPRI	Indicates the priority for deployment of the ZTIMEQ object, relative to the other elements deployed during the Client Connect. Elements with priority numbers lower than the value of ZOBJPRI are deployed before this ZTIMEQ object. A value of 90 is inherited from the BASE INSTANCE.
ZSCHMODE	Specifies the timer owner. Leave as Default.
ZSVCOID	Specifies the object ID of the Application instance that this Scheduling instance is connected to. The value is inherited from the BASE INSTANCE.

Table 8.6 ~ Scheduling (TIMER) Attributes

Attribute	Usage
ZCHNNAME	Specifies the name of the domain in the Radia Database where the Application instance to which this Scheduling instance is connected. The value is inherited from the BASE INSTANCE.
ZPRVNAME	The name of the Radia Configuration Server that the subscriber receiving this timer instance is connected to. The value is inherited from the BASE INSTANCE.
ZCREATE	The Scheduler Create method that runs on the client computer. The value is inherited from the BASE INSTANCE.
ZVERIFY	The Scheduler Verify method that runs on the client computer. The value is inherited from the BASE INSTANCE.
ZUPDATE	The Scheduler Update method that runs on the client computer. The value is inherited from the BASE INSTANCE.
ZDELETE	The Scheduler Delete method that runs on the client computer. The value is inherited from the BASE INSTANCE.

Configuring the Timer

Now that you are familiar with the attributes in the timer instance, we will review the syntax used to configure these attributes. Then, we will configure the sample timer using the appropriate values to deploy mandatory applications every week at random intervals 5:00 PM and 7:00 PM.

Specifying When the Timer Expires (ZSCHDEF)

Use the ZSCHDEF and ZSCHFREQ attributes respectively to specify when the timer should expire, and how often. ZSCHDEF indicates when the timer should expire and ZSCHFREQ indicates how often the timer should expire.

If you want your timer to expire only one time (ZSCHFREQ = ONCE), or repeatedly (ZSCHFREQ = PERIODIC) at a certain time, use the following table to determine the appropriate syntax for the value of ZSCHDEF.

Table 8.7 ~ Syntax of ZSCHDEF Attribute

Type	Syntax	Timer Expires
DAILY	DAILY(&ZSYSDATE,24:00:00)	Daily at midnight on system's date.
WEEKLY	WEEKLY(&ZSYSDATE,01:00:00)	Every 7 days at 1:00 AM.
WEEKDAY	WEEKDAY(Name of Weekday*,01:00:00)	Every *Name of Weekday** at 1:00 AM. The weekday must be specified in uppercase. *** *Name of Weekday* is the name of a specific weekday, e.g. Monday.**
HOURLY	HOURLY(&ZSYSDATE,08:41:00)	Hourly starting at 8:41 AM on system's date.

Table 8.7 ~ Syntax of ZSCHDEF Attribute

Type	Syntax	Timer Expires
INTERVAL	INTERVAL(&ZSYSDATE,08:41:00,,30)	Every 30 minutes starting at 8:41 AM based on system's date. **Note**: When setting ZSCHFREQ to RANDOM, there are four possible arguments for INTERVAL. The extra comma is required whether ZCHFREQ is set to RANDOM or not. Use the second argument to set the start time and the third argument to set the end time of the random interval.
MONTHDAY	MONTHDAY(*Name of Weekday**,01:00:00,,2)	Every *Name of Weekday** at 1:00 AM on the second week of the month. The weekday must be specified in uppercase. If the last argument is not specified, then the timer will run on the first week of the month. **Note**: When setting ZSCHFREQ to RANDOM, there are four possible arguments for MONTHDAY. The extra comma is required whether ZCHFREQ is set to RANDOM or not. Use the second argument to set the start time and the third argument to set the end time of the random interval.
MONTHLY	MONTHLY(20040215,01:00)	Runs on the 15th of the month starting in February at 1:00 AM. If setting ZSCHFREQ to RANDOM, use the second argument to set the start time and add a third argument to set the end time of the random interval. **Note**: MONTHLY reschedules differently than other schedule frequencies MONTHLY will reschedule from the original day it was scheduled for, instead of the day it ran. For example, if ZSCHDEF was MONTHLY(20040116,05:30:00) and the client device was off on the 16th of January, and didn't execute until the 18th of January, the new schedule would be MONTHLY(20040216,05:30:00) instead of MONTHLY(20040118,05:30:00).
NUMDAYS	NUMDAYS(20000803,08:00:00,,14)	Every 14 days starting on August 3, 2000 at 8:00 AM.
STARTUP	STARTUP	When the Radia Scheduler starts on the client device, it will immediately execute all Timer instances with ZSCHDEF of STARTUP. It will check for all special conditions (NETAVAIL, ZNOPING and RETRYFLG). After it executes all the STARTUP instances, RADSCHED then goes back into its regular timer loop. It will only execute STARTUP instances in the regular timer loop if the ZPENDING flag on that instance was set (because NETAVAIL or ZNOPING could not get through or RETRYFLG is on and the return code was 200 during startup run).

DEPLOYING APPLICATIONS OVER A PERIOD OF TIME

You can also deploy applications over a specified range of time. This varies the load on the Radia Configuration Server and alleviates network congestion by spreading out the deployment over a period.

To do this, indicate that you want your timer to expire randomly (ZSCHFREQ = RANDOM). Then, use ZSCHDEF to specify the range of time over which the applications should be deployed.

To do this, the syntax for ZSCHDEF is:

```
ZSCHDEF =<frequency>(<date>, <from_time>, <to_time>, <limit>)
```

Table 8.8 ~ ZSCHDEF Parameters when ZSCHFREQ is Set to RANDOM	
Parameter	**Description**
<frequency>	Any of the supported frequency values, including DAILY, HOURLY, INTERVAL, NUMDAY, WEEKLY, and WEEKDAY.
<date>	Date when the event should be initiated. The format is YYYYMMDD.
<from_time>	Beginning time for randomization. The format is HH:MM:SS.
<to_time>	Ending time for randomization. The format is HH:MM:SS.
<limit>	The optional parameter that prevents initiation after this time (HH:MM:SS). The format is HH:MM:SS.

In our example, we want to configure the timer to deploy mandatory applications on a weekly basis. However, to alleviate network congestion, we are going to schedule deployments to run between 5:00 PM and 7:00 PM.

To specify when the timer expires

1. After navigating to the timer instance, double-click **ZSCHFREQ** in the list view of the Radia System Explorer.

   The **Editing Instance** dialog box opens.

2. In the **Frequency** drop-down list, select **RANDOM**.

3. Click **ZSCHDEF**.

4. In the **Timer Parameter** text box, type `WEEKLY(&ZSYSDATE,17:00:00,19:00:00)`.
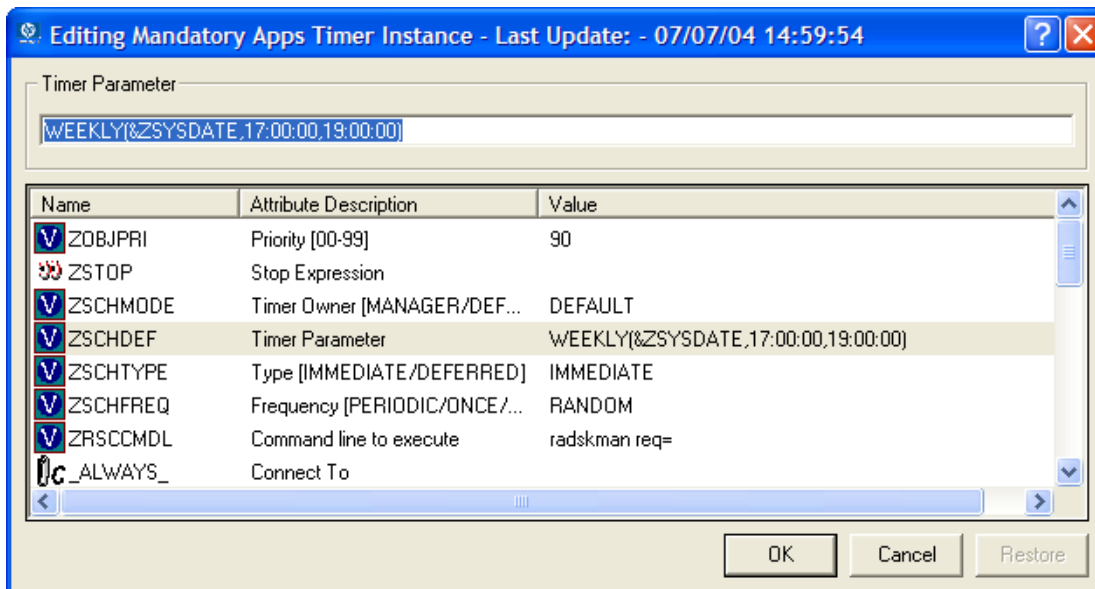
Figure 8.6 ~ Specify value for ZSCHDEF.

5. Click **ZSCHTYPE**.

6. In the **Type (Immediate/Deferred)** drop-down list, select **IMMEDIATE**.

7. If you are done editing the attributes for the timer instance, click **OK**, and then click **Yes** when you are prompted to confirm your changes.

8. **OR**

9. Select the next attribute to edit.

Specifying the Command Line (ZRSCCMDL)

When the timer expires, it executes any command line that you specify on the client computer.

> **Note**
>
> If you want to see how timers work, you might create a timer that runs a command line such as **SystemDrive:\Notepad.exe**. Remember to configure the timer to expire immediately, and attach it to a service. Then, deploy the service. When the timer expires on the client computer, the Notepad application opens.

To specify a command line

1. Navigate to the timer instance and then double-click **ZRSCCMDL** in the list view of the Radia System Explorer.

   The **Editing Instance** dialog box opens.
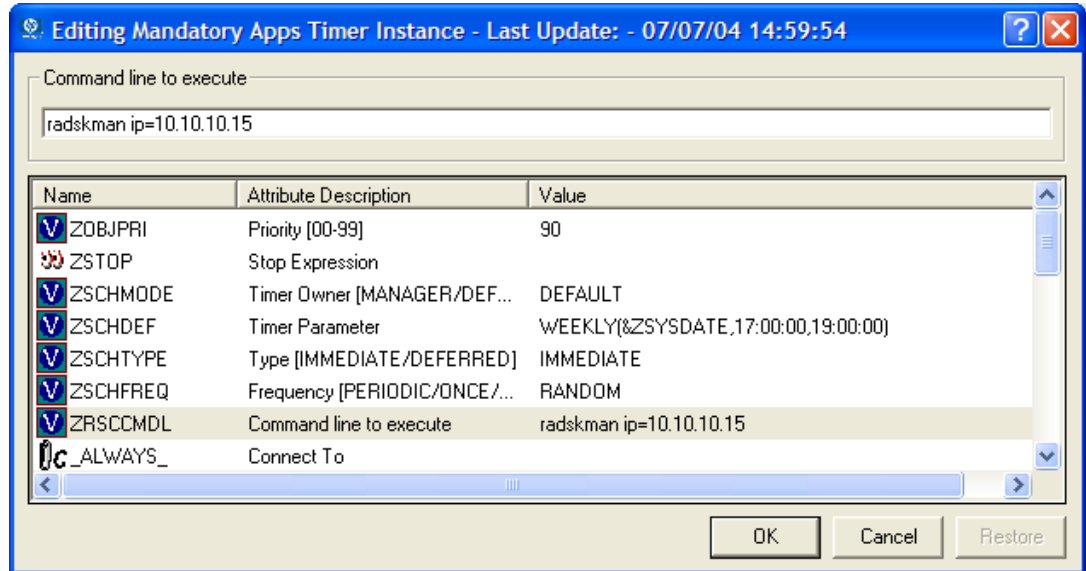
Figure 8.7 ~ Specify the command line.

2. In the **Command line to execute** text box, type the command line to execute the appropriate program for your needs.

3. Click **OK**.

4. Click **Yes** when you are prompted to confirm your changes.

In our example, we indicated that we would be deploying new mandatory applications to your subscribers on a weekly basis. The following procedure will show you how to specify a command line that will update *all* mandatory services and perform client self-maintenance.

Connecting the Timer to a Service

Once you have created the timer, you must connect it to a service. Each subscriber that receives the service to which the timer is connected will receive the timer information in the ZTIMEQ object the next time his Radia Client connects to the Radia Configuration Server.

In the example we have used throughout this section, we created a timer intended to deliver mandatory applications. We will connect the sample timer to the GS-Calc service. We assume that all subscribers are receiving this service.

1. To connect the timer to the GS-Calc service

2. In the Radia System Explorer, double-click **PRIMARY**.

3. Double-click **SOFTWARE**.

4. Double-click the **Application (ZSERVICE)** class.

5. Right-click GS-CALC. A menu opens.

6. Select **Show Connections**. The **SOFTWARE.ZSERVICE Connections** dialog box opens.
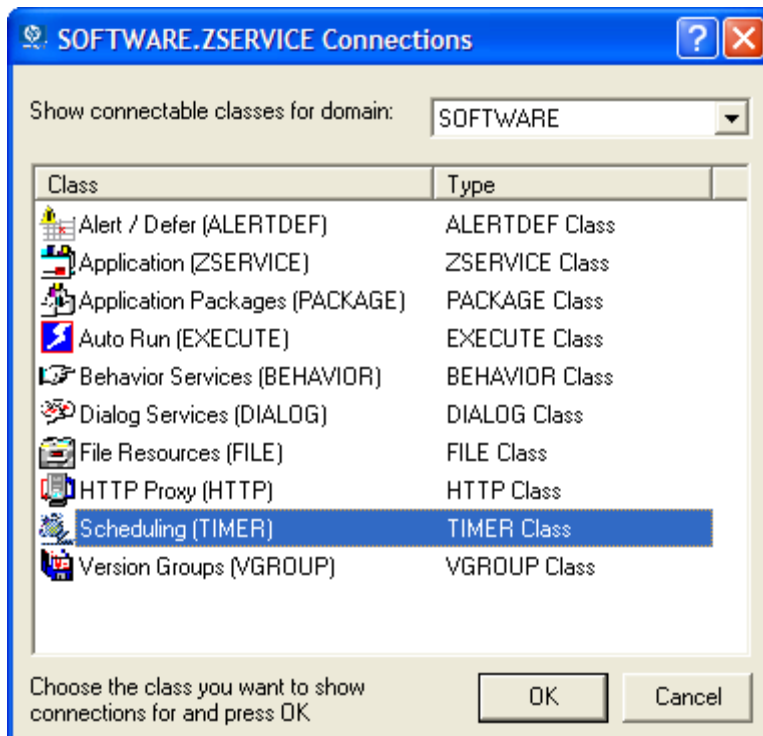
Figure 8.8 ~ Show Connectable Classes dialog box.

7. Select **Scheduling (TIMER)**, and then click **OK**. The Timer class instances appear in the list view.
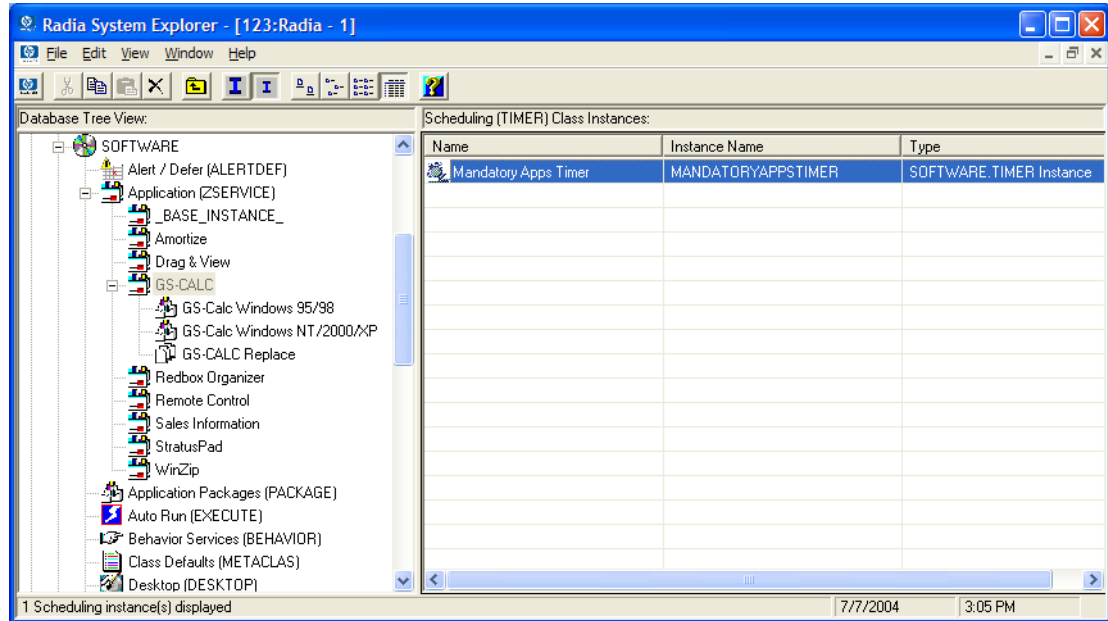
Figure 8.9 ~ Timer class instances in the list view.

8. Click **Mandatory Apps Timer** in the list view and drag it to the appropriate Application (ZSERVICE) instance (in this example, **GS-Calc**). When your cursor changes to a paper clip, release the mouse button. The **Select Connection Attribute** dialog box opens.

9. Click **Copy**.

10. Click **Yes** to confirm that you want to connect the **GS-Clac** service to the **Mandatory Apps Timer**.

11. Click **OK** to close the confirmation message.

Testing the Timer Deployment

The first time a client computer connects to the Radia Configuration Server after the timer is created, the timer information is transferred to the client computer in the ZTIMEQ object. The ZTIMEQ object is located in the IDMROOT directory (by default, *SystemDrive***:\Program Files\Novadigm\Lib**).

In this section, we will force the client computer to connect to the Radia Configuration Server so that you can view the ZTIMEQ object. For testing purposes, we will run a command line on the client computer that updates *all* mandatory services and performs any client self-maintenance.

To connect to the Radia Configuration Server

1. On the client computer, go to a command prompt and change the directory to *SystemDrive*:**Program Files\Novadigm**. This is the default location for **radskman**.

2. Type

    ```
    radskman ip=<manager ip>,port=<mgr_port>
    ```

    See *Connection Parameters (Radskman)* on page 345 for information about **radskman** and the parameters above.

    > **Note**
    >
    > If you plan to do further testing, consider creating a batch file that contains the command line. Save the file in IDMSYS (by default *SystemDrive*:**Program Files\Novadigm**) on the client computer. Then, create a shortcut on the desktop of the client computer.

3. Press the ENTER key on your keyboard. Once the Client Connect is finished, you can view the ZTIMEQ object on the client computer.

Viewing the Timer Object (ZTIMEQ.EDM)

Now that we have forced the client computer to connect to the Radia Configuration Server, the ZTIMEQ object is stored on client computer. Use the Radia Client Explorer, installed as part of the Radia Administrator Workstation, to view or modify the ZTIMEQ object.

> **Note**
>
> After the timer expires, the ZTIMEQ object is removed from the client computer during the next Client Connect.
>
> If the timer is configured to expire only one time, in the TIMER.ZSCHFREQ attribute, it will be removed immediately after the timer expires, during the next Client Connect. If the timer is configured to expire more than one time, the ZTIMEQ object will be removed after the timer expires for the last time, during the next Client Connect.

The ZTIMEQ object contains one instance for each Scheduling (TIMER) instance in the Radia Database. For example, if two different services each have timer instances associated with them, then there will be two instances in the ZTIMEQ object.

To view the ZTIMEQ object on the client computer

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia Client Explorer**.
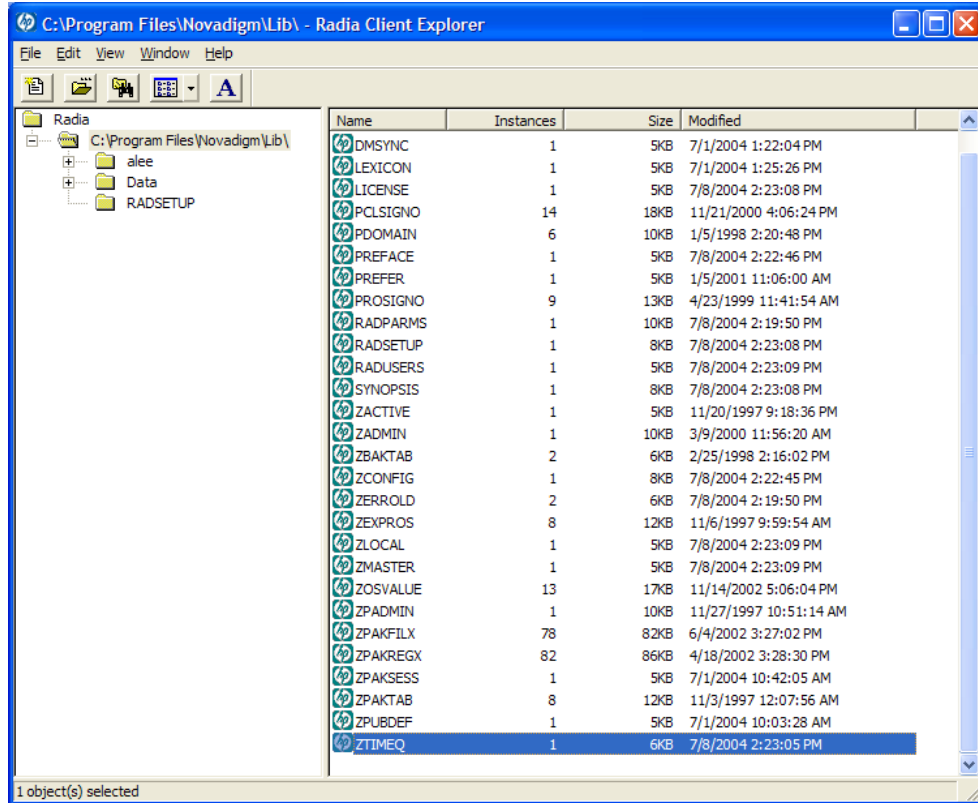
Figure 8.10 ~ Radia Client Explorer objects.

2. Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.
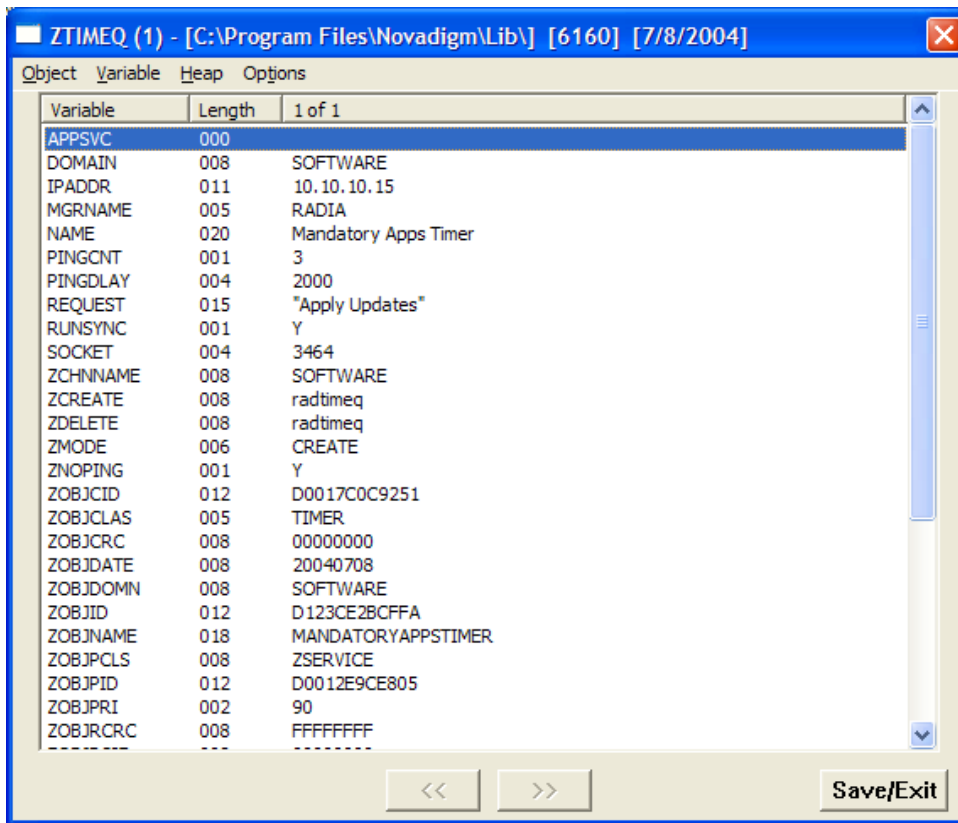
Figure 8.11 ~ An instance of the ZTIMEQ object.

Experimenting with Timers

If you want to experiment with timers, you can modify the ZRSCCMDL, ZSCHDEF, ZSCHFREQ, and ZSCHTYPE attributes in the ZTIMEQ object on the client computer to see what happens in various situations.

To edit an attribute in ZTIMEQ

1. Double-click the attribute that you want to edit.

The **Change Variable** dialog box opens.

2. Type the new value.

3. Click **Save/Exit**.

To quickly determine whether the timer expires, you can change ZRSCCMDL to run any executable, such as Notepad. When the timer expires, Notepad opens, confirming that the timer expired.

Timer Logs

Timer events are tracked in three logs, stored in the IDMLOG directory (by default, *SystemDrive***:\Program Files\Novadigm\Log**).

The following table describes the timer logs.

Table 8.9 ~ Timer Logs	
Log File	**Usage**
RADSCHED.LOG	Lists the results of the most recent Scheduler expiration.
	The Scheduler, **radsched**, runs in the background. It wakes up once a minute and examines the ZTIMEQ client object to see if a timer has expired. This log only retains information from the most recent expiration.
RADSHIST.LOG	Lists all of the programs dispatched because a timer instance expired. It reflects all activity since **radsched** was started last.
RADTIMEQ.LOG	Lists the events that occurred during the last execution of the **radtimeq** method.
	This method executes when the application to which the timer is attached is created, updated, verified, or deleted. Only the last execution's events appear in the log, with an indication of what activity took place regarding the application.

Notifying Subscribers

Use Radia Notify to force one or more client computers to connect to the Radia Configuration Server to install, update, or remove an application. Each client computer runs the Radia Notify service in the background. This service waits to receive a Notify message from the Radia Configuration Server. When a message is received, the client computer connects to the Radia Configuration Server and performs the action initiated by the Notify operation. Radia Notify can also send e-mail notification to client computers.

Note
If you are using the Radia Management Portal, you can use the Notify Task. See the *Radia Management Portal Guide* for details.

You can initiate a Notify by:

■ Selecting **Notify Subscribers** from the shortcut menu for an Application (ZSERVICE) instance. *Only* use this option to **update** or **remove** applications. You cannot use this type of Notify to install an application because this option notifies *existing* subscribers.

■ Creating a Drag-and-Drop Notify command. Use this option to install, update, or remove an application. The benefit of this type of Notify is that the application does not have to be installed on the client computer to perform the Notify.

Caution
Drag-and-Drop Notify is intended for use in environments with a single Radia Configuration Server. If you are working in an environment with multiple Radia Configuration Servers, consider using the Radia Management Portal. Contact your sales representative for details.

Requirements for Using Notify

To use Notify:

■ The client computer must connect to the Radia Configuration Server prior to the notification. This populates the PROFILE file, which contains the client computer's network address, used by Notify.

> **Caution**
>
> Notify is designed to notify only subscribers whose information is in the PROFILE file in the Radia Database.

■ Confirm that the Radia Configuration Server Settings file, EDMPROF.DAT, is configured properly, as shown in Figure 8.12 below. This file and the relevant lines are created in the Radia Configuration Server Settings file when the Server is installed.

```
[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
VERIFY_INTERVAL = 5
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexxmgr) RESTART=YES
CMD_LINE=(ztcpmgr PORT=3464,NAME=tcpmgr_3464) RESTART=YES
CMD_LINE=(znfytmgr NAME=NotifyManager) RESTART=YES  ⬅ Line needed for Notify Manager
CMD_LINE=(zrtrymgr) RESTART=YES
```

Figure 8.12 ~ Radia Configuration Server Settings file for Notify.

■ If you are using Drag-and-Drop Notify to run a command, you must store the program that you want to execute in the IDMSYS directory (by default *SystemDrive*:**\Program Files\Novadigm**).

■ If you are using e-mail to notify subscribers, be sure that the correct e-mail address for the subscriber is stored in the EMAIL attribute of the USER instance in the USER class in the POLICY domain.

■ If you are using e-mail to notify subscribers, be sure that the Radia Configuration Server is properly configured for email.

Initiating a Notify from a ZSERVICE Instance

To update or remove a service, initiate the Notify from the shortcut menu for the Application (ZSERVICE) instance.

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.

To initiate a Notify from a ZSERVICE Instance

1. Right-click the **Application (ZSERVICE)** instance, such as **GS-CALC**.

A shortcut menu opens.
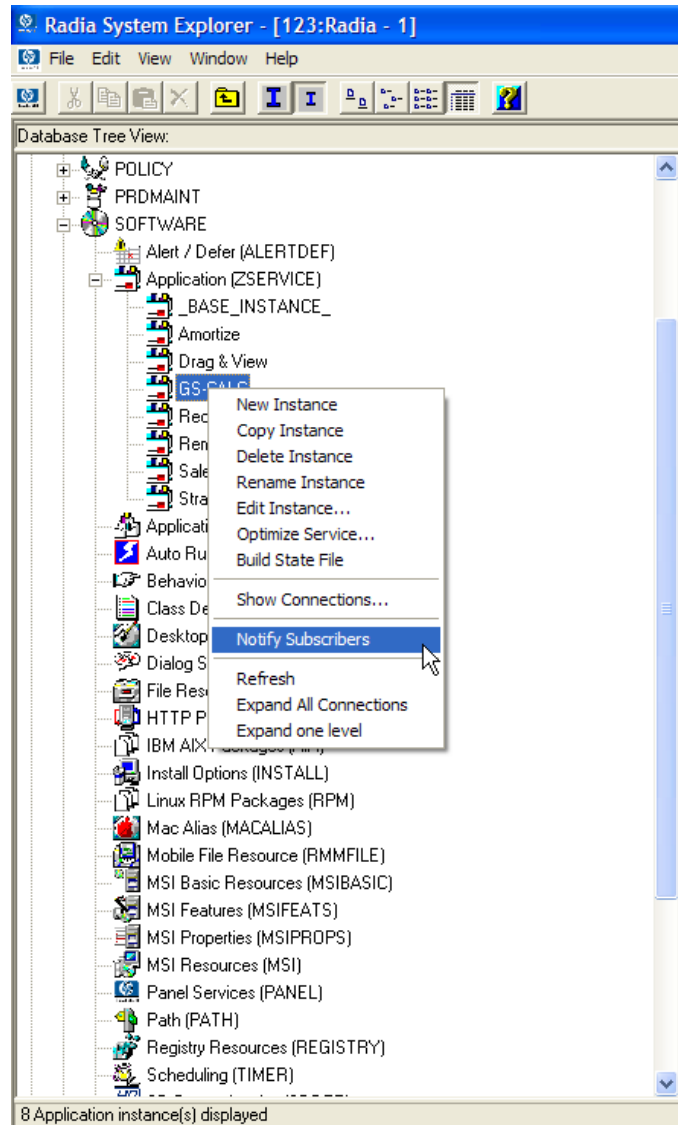
Figure 8.13 ~ Application (ZSERVICE) instance shortcut menu.

2. Select **Notify Subscribers**.

The Notify retrieves the list of the subscribers from the POLICY domain. If the selected application does not have any subscribers, the following message appears.
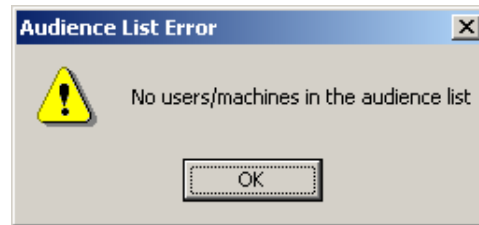
Figure 8.14 ~ Alert message when no users are in audience list.

If the selected application does have subscribers, a message appears to confirm that you want to build an audience list.

3. Click **Yes** to confirm that you want to build an audience list.

The **Radia Notify Manager** opens.
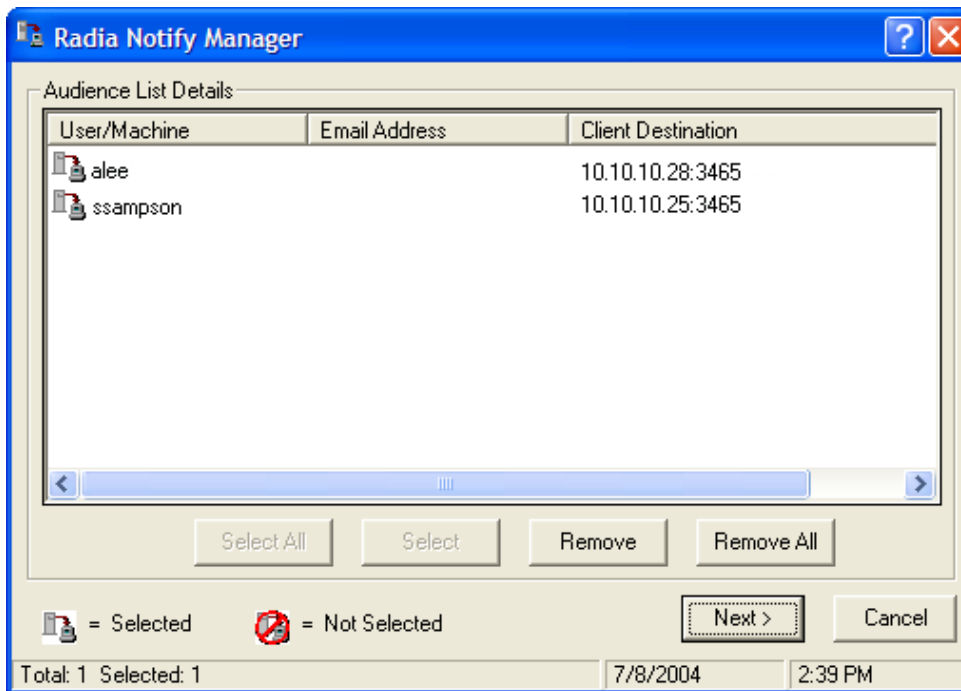
Figure 8.15 ~ Notify audience list.

A list of the subscribers to the application appears in the **Audience List Details** area.

● By default, the *entire audience* will be notified.

● To select individual subscribers, click **Remove All**. Then, select the appropriate subscribers and click **Select**.

● To remove an individual subscriber, select the appropriate subscriber, and then click **Remove**.

As shown in the figure above, the symbols to the left of the subscriber indicate who has been selected or not. The total number of client computers in the audience list and the number of client computers selected are displayed at the bottom left of the dialog box.

4. Click **Next** when you are done selecting client computers in the Notify audience list.

5. Select the **Notification Type** for all members of the audience list.

● Send an Email

Select this option to inform subscribers of an application's status. When using the Radia Dual Mode installation, you can also use this option to inform subscribers of an application's optional availability. See the *Radia Software Manager Guide for Windows* for more information.

The subscriber's e-mail attribute (EMAIL) in the user instance *must* contain a valid entry. Be sure to complete the **Subject** and **Message** fields.

● Update the Application on the target machine(s)

Select this option to install updates or new versions of an application on the client computers.

● Remove the Application on the target machine(s)

Select this option to remove an application from the client computers.

Normally, Notify removes the application without requesting permission from the client. This allows removal of applications from unattended client computers.

If you want to require the subscriber to give permission to remove an application, select the **Prompt for deletion on client** check box.
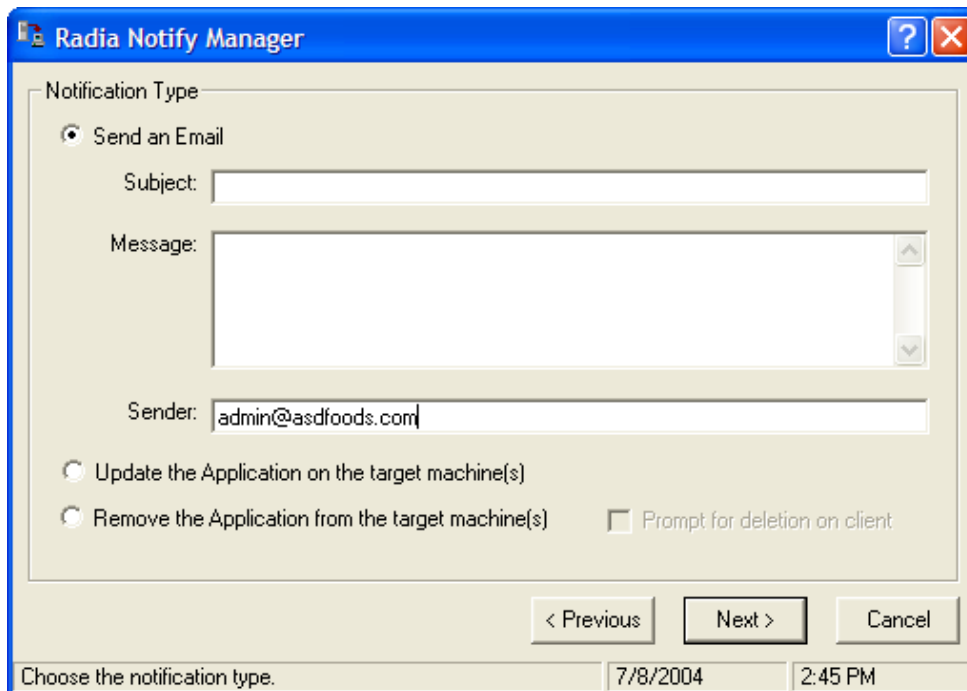
Figure 8.16 ~ Radia Notify Manager Notification Type.

6. Click **Next** to continue.

   The **Notification Details** dialog box opens.

   By default, the Notify occurs immediately and generates an object in the NOTIFY file in the Radia Database. The object is named according to the date and time of the Notify action in the following format: YYYY_MM_DD_HH_MM_SS.

   Use this dialog box to create a custom domain to store the object or to defer the notify action to a later date and time.

   ● Select **Use Custom Notify Domain** and type a name in the text box for the new domain, located in the NOTIFY file, in the Radia Database.

   ● Select **Deferred Notify** and use the **Time (hh/mm)** drop-down lists and the calendar controls to schedule the Notify.
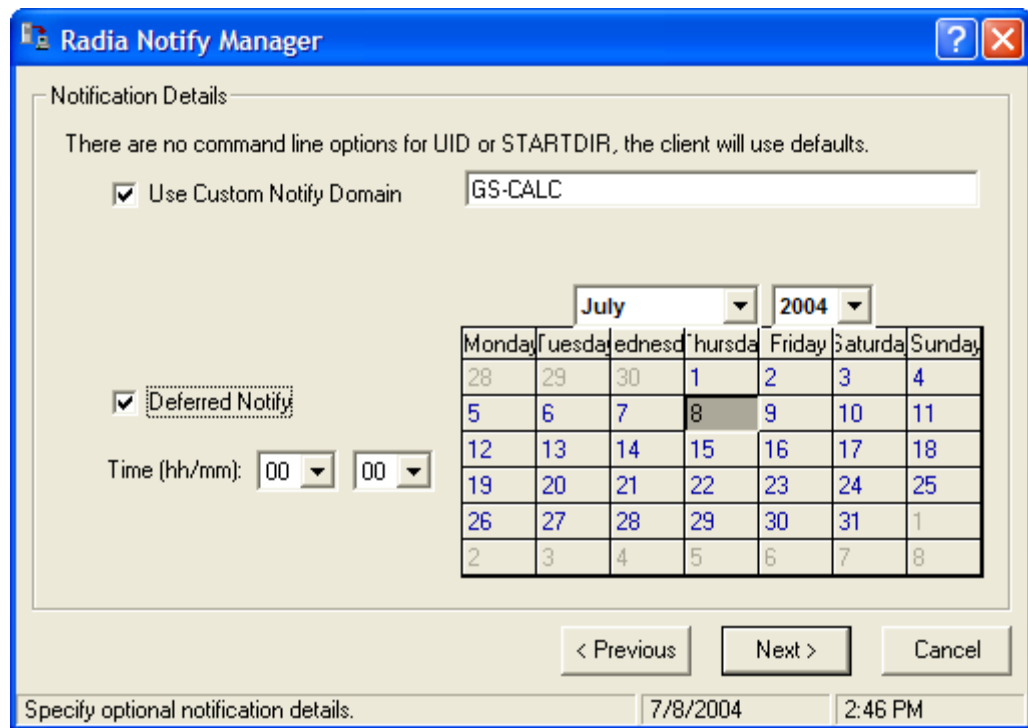
Figure 8.17 ~ Notification Details dialog box.

7. Click **Next**.

The **Notification Summary** dialog box opens.
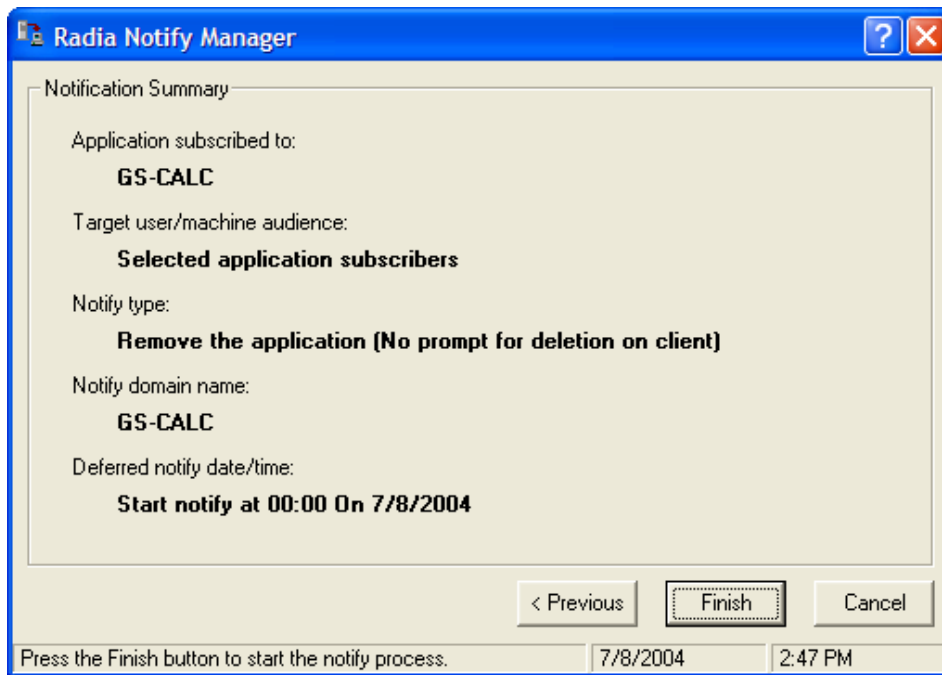
Figure 8.18 ~ Notification Summary dialog box.

8. Click **Finish** to begin the Notify.

A message asks if you want to start the status monitor.
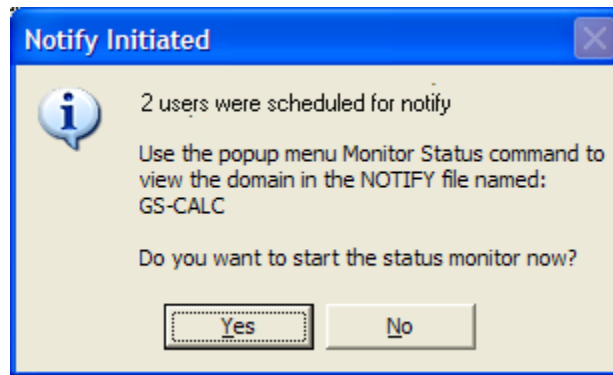
Figure 8.19 ~ E-mail Notify initiated.

9. Click **Yes** to view the status of the Notify.

A dialog box opens with a list of the subscribers and the status of the notification.
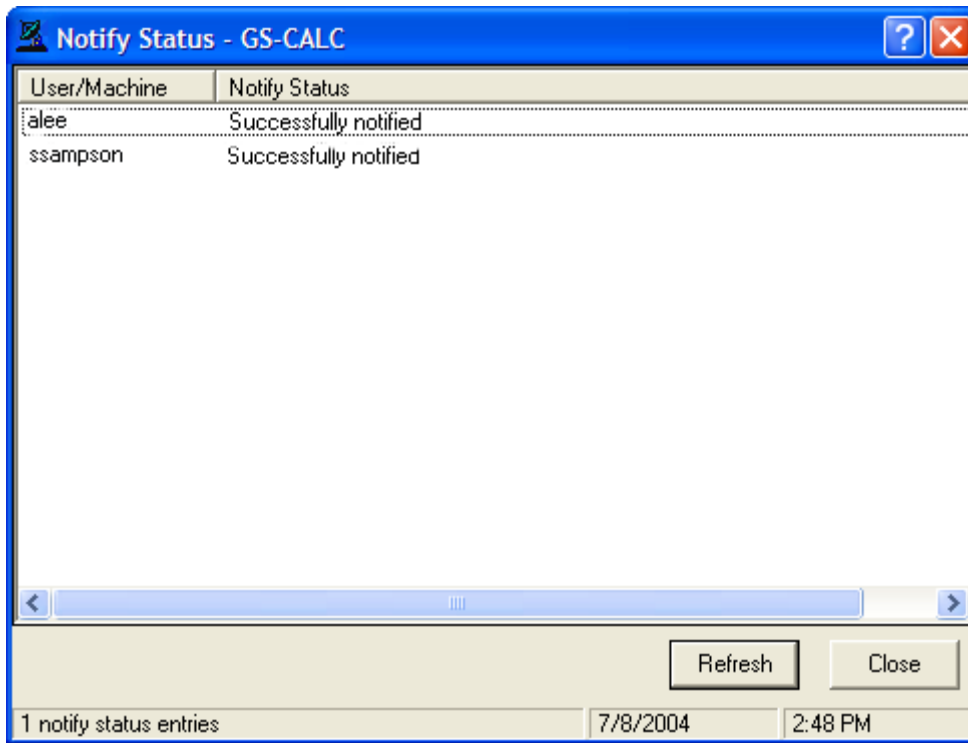
Figure 8.20 ~ Notify Status Monitor dialog box.

10. Click **Refresh** to update the Status Monitor.

11. Click **Close** when you are done.

Creating a Drag-and-Drop Notify Command

Use a Drag-and-Drop Notify command to initiate a Notify to one or more subscribers immediately. The benefit of this type of Notify is that the application does not have to be installed on the client computer to perform the Notify. You might use this type of notify to install software, update all mandatory services, or even run an executable on the client computer.

The Drag-and-Drop Notify works only if the client computer has connected to the Radia Configuration Server prior to the notification. This populates the PROFILE file, which contains the client computer's network address, used by Notify.

Caution
The Drag-and-Drop Notify is intended for use in environments with a single Radia Configuration Server. If you are working in an environment with multiple Radia Configuration Servers, consider using the Push Manager. Contact your sales representative for details.

In the following example, we will create a command that will update all mandatory services on your client computers.

To create a Drag-and-Drop Notify

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

Note
The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**. The **Radia System Explorer** window opens.
3. Double-click **PRIMARY**.
4. Double-click **SYSTEM**.
5. Double-click **Application Manager (ZCOMMAND)**.
6. Right-click the **Mandatory** instance, and select **Copy Instance**.

   The **Copy Instance** dialog box opens.
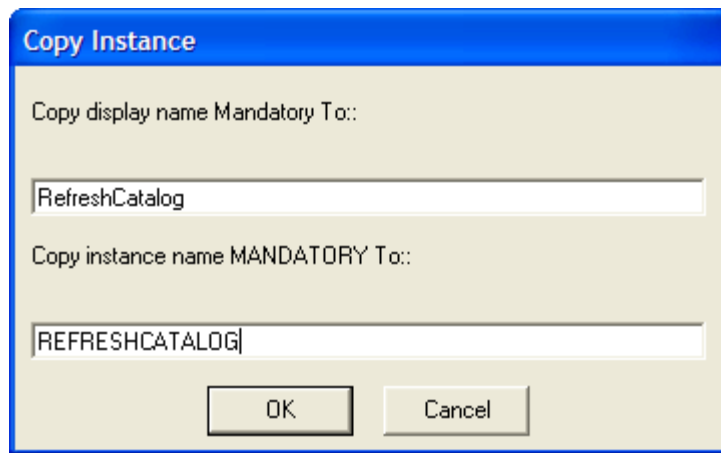
Figure 8.21 ~ Copy Mandatory instance to RefreshCatalog instance.

7. Type a display name and name for the instance in the appropriate text boxes. For this example, we named the instances **RefreshCatalog**.

8. Click **OK**.

The **RefreshCatalog** instance appears in the list of ZCOMMAND class instances.

9. Double-click the instance, such as **RefreshCatalog**, in the tree view.

The attributes appear in the list view.

10. Double-click the **ZCMDPRMS** attribute.

The **Edit Instance** dialog box opens.
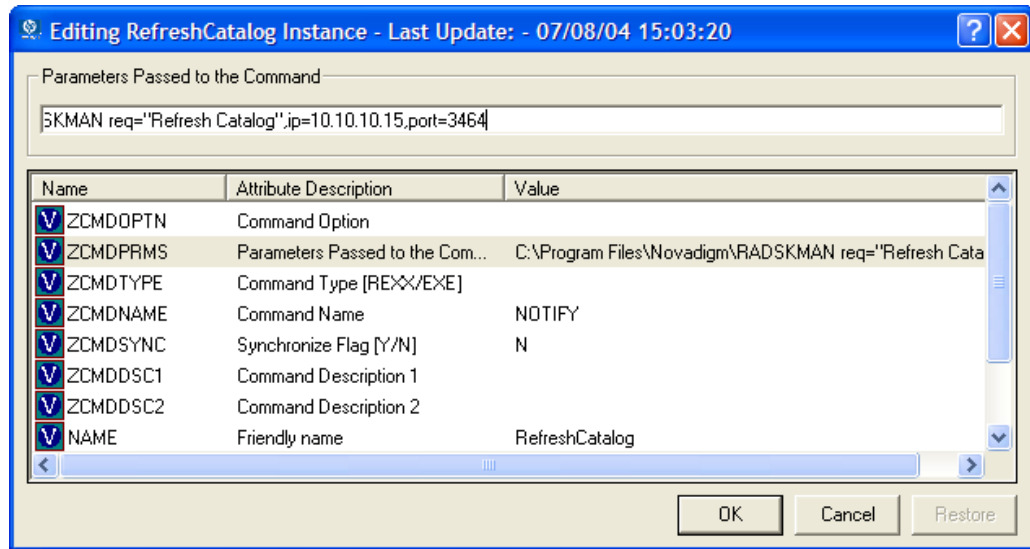
Figure 8.22 ~ Editing the ZCMDPRMS attribute in the RefreshCatalog instance.

11. Type the command line that you want to execute on the client computer. For this example, we will type:

```
radskman ip=<mgr_ip>,port=<mgr_port>
```

This command line updates or installs *all* new and old mandatory applications. See *Specifying the Command Line* on page 365 for more information.

Caution
To use a Drag-and-Drop Notify to run a command, you must store the program that you want to execute in the IDMSYS directory (by default ***SystemDrive*:\Program Files \Novadigm**).

12. Click **OK**.

13. Click **Yes** to confirm that you want to save your changes.

14. From the **POLICY** domain, select a User, Workgroup, or Department instance and drag it to the **RefreshCatalog** command. The cursor changes to a wand.

15. Release the mouse button.

The Notify is sent immediately to the specified subscribers and the command line in ZCMDPRMS is executed.

Retrying a Notify

Sometimes a subscriber cannot be notified. This may occur for one of the following reasons:

■ The client computer may be turned off.

■ The subscriber does not have a valid e-mail address listed in the Radia Database.

■ The client computer is not running the Radia Notify service.

■ The client computer may not be accessible via the normal communication channel.

An unsuccessful Notify attempt creates an instance in the RETRY domain of the NOTIFY file. The RETRY domain is created the first time a Notify fails.
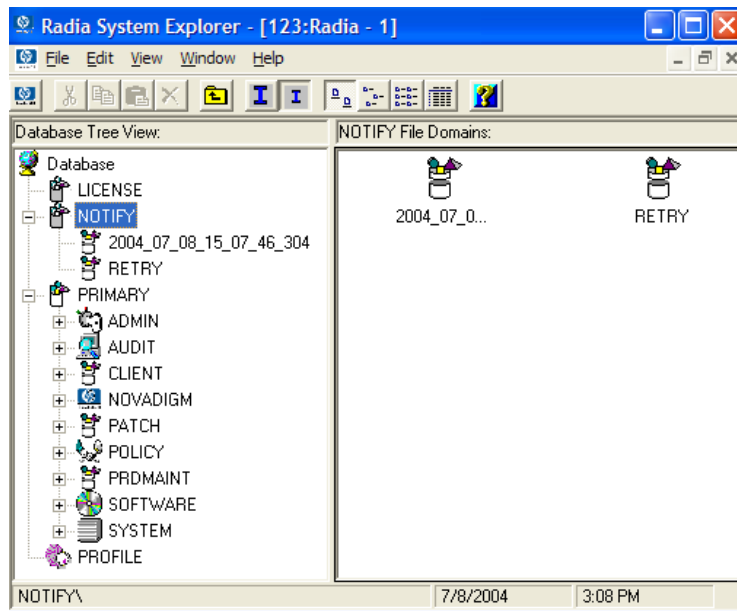
Figure 8.23 ~ RETRY domain.

By default, Radia automatically retries the Notify operation for failed attempts. To do this, the Radia Configuration Server is started with the Notify Retry Manager (**zrtrymgr** module), as indicated in the following excerpt from the Radia Configuration Server Settings file, EDMPROF.DAT, located in the bin directory of your Radia Configuration Server's installation directory (by default, **C:\Radia\ConfigurationServer\bin**).

```
[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
VERIFY_INTERVAL = 5
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexxmgr) RESTART=YES
CMD_LINE=(ztcpmgr PORT=3464,NAME=tcpmgr_3464) RESTART=YES
CMD_LINE=(znfytmgr NAME=NotifyManager) RESTART=YES
CMD_LINE=(zrtrymgr) RESTART=YES
```
⟵ **Line needed for Retry Manager**

Figure 8.24 ~ Line to include in the Radia Configuration Server Settings file to enable Notify Retry.

Caution

If you make any changes to the Radia Configuration Server Settings file, EDMPROF.DAT, you must restart the Radia Configuration Server service.

For more information on editing the Radia Configuration Server Settings file, see the *Radia Configuration Server Guide* (previously known as the *Manager Guide*).

The Notify Retry Manager periodically examines the NOTIFY file's RETRY domain, based on VERIFY_INTERVAL in EDMPROF.DAT. The default interval is every five minutes. The Retry Manager attempts the Notify operation for each instance it finds in the RETRY domain.
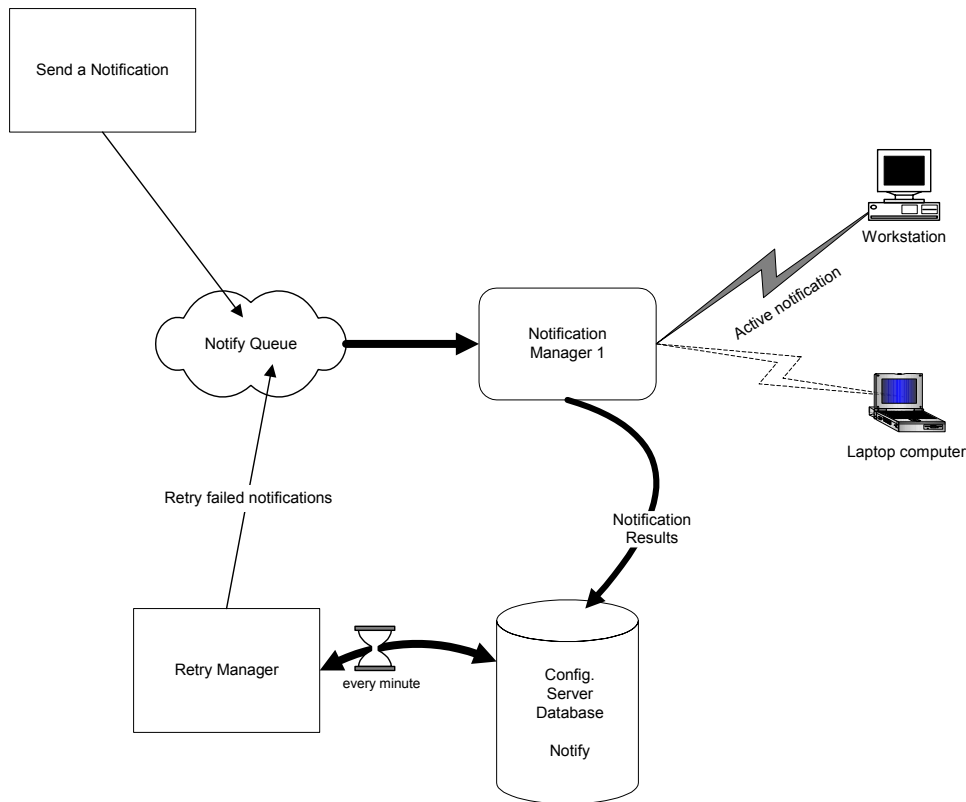
Figure 8.25 ~ Retry process.

Viewing the Results of a Notify

You can find information about a Notify:

■   In the Radia System Explorer in the NOTIFY file.

■   In the Status Monitor accessed from the NOTIFY file in the Radia System Explorer.

You can also find information about a Notify operation in the Radia Configuration Server log. The log file is stored on the Radia Configuration Server in the LOG directory (by default, **C:\Radia\ConfigurationServer\log**).

Viewing an Instance in the NOTIFY file

The NOTIFY file is created after the first Notify is initiated. Each Notify operation creates a single object in the NOTIFY file in the Radia Database. The objects are named according to the date and time of the Notify action in the following format: YYYY_MM_DD_HH_MM_SS. In each

object, there is an instance for each subscriber that was notified. Each instance contains important information about the subscriber and the notify operation.

To view an instance in the NOTIFY file

1. In the Radia System Explorer, double-click **NOTIFY**.
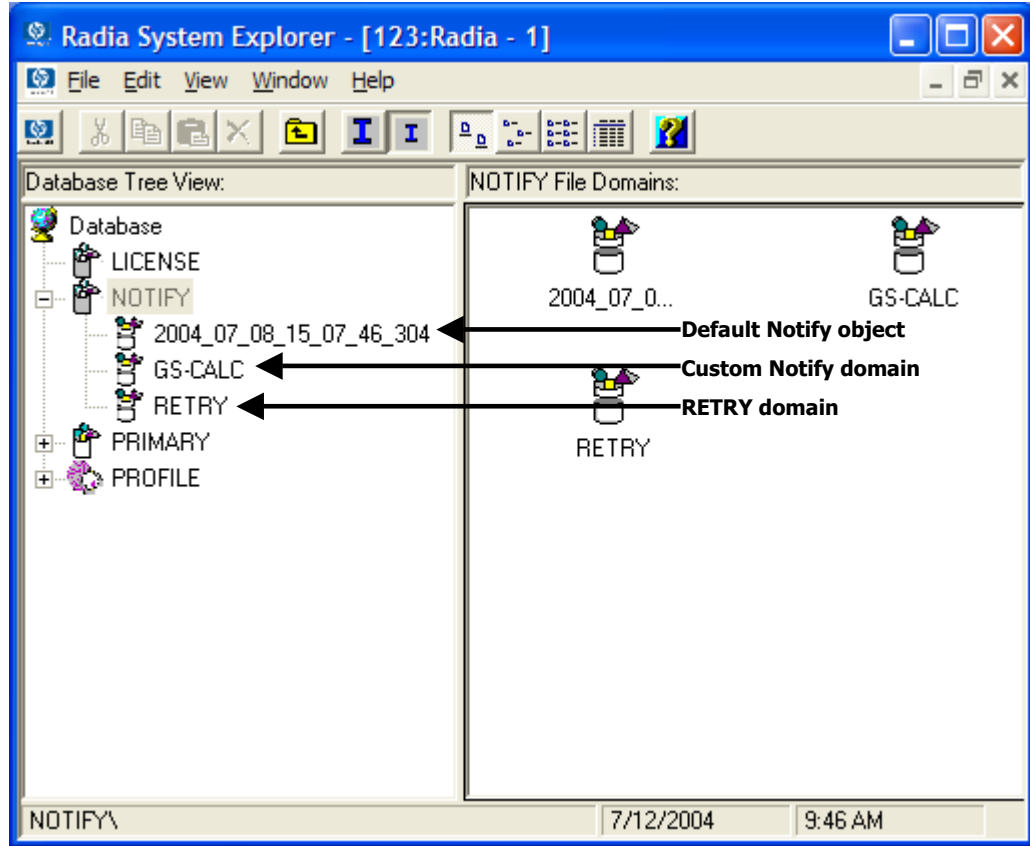
Figure 8.26 ~ NOTIFY file.

Notice in the image above, there are several default Notify objects. Each object represents a single Notify operation. The objects are named according to the date and time of the Notify action in the following format: YYYY_MM_DD_HH_MM_SS.

The custom Notify domain, REDBOX, is also a Notify object. However, this Notify was given a custom domain name in the **Notification Details** dialog box.

Finally, notice the RETRY domain. An unsuccessful Notify attempt creates an instance in the RETRY domain of the NOTIFY file. The RETRY domain is created the first time a Notify fails.

2. Double-click the Notify object that you want to review.

3. Double-click **NOTIFY**.

The NOTIFY file is divided into domains, where each domain represents one Notify operation. The name of the domain is in the form YYYY_MM_DD_HH_MM_SS, representing the date and time when the Notify operation was initiated.

Each NOTIFY domain has one NOTIFY class. Each NOTIFY class contains an instance for each subscriber that was notified. The instances are named with eight-digit numbers starting with 00000001 and running sequentially up to the total number of notified subscribers.

Each instance contains attributes that identify the subscriber, the kind of Notify operation, and the results of the Notify operation for that subscriber.

4. Review the attributes in the list view. The following table describes possible attributes.

Table 8.10 ~ Attributes in the NOTIFY Instance	
Attribute	**Description**
ZUSERID	The USER, WORKGRP or DEPT that you notified.
ZCIPADDR	The IP address of the client computer.
EMAIL	The subscriber's e-mail address, if using e-mail notification.
NTFYTYPE	Indicates the type of notify, such as E for e-mail notification.
NTFYDATE	The date of the Notify.
NTFYTIME	The time of the Notify.
NTFYMSG	Message indicating the status of the Notify, such as "Successfully notified."
NTFYRC	The return code generated for a Notify.
NTFYCMDL	The command line that the Notify executed.
NTFYSUBJ	The subject of the email that is sent, if using e-mail notification.
LOCALUID	The user ID for the subscriber that is currently logged on to the computer.
NTFYRTIM	The time at which the notification should execute.
NTFYRNUM	The number of times to retry the Notify.
NTFYDOMN	The name of the domain where this instance is stored.
NTFYINS	The name of this instance.
NTFYPORT	The registered port for Notify.
NTFYPWD	The encrypted password for the Notify.
NTFYUINF	The user information passed to the Notify operation from the Notify Manager.
NTYFYRMAX	The maximum number of times to retry the Notify.
NTYFYDLAY	The amount of time (in seconds) to wait before retrying the Notify.

Table 8.10 ~ Attributes in the NOTIFY Instance	
Attribute	**Description**
NTYFYMAC	The physical address of the client computer. Used for Wake-on-LAN support.
NTYFYMASK	The network mask used for Wake-on-LAN support.

Viewing Results of a Notify or Retry in the Status Monitor

Use the Status Monitor to review the results of a Notify or Retry operation for all of your subscribers.

To see the status of a Notify or Retry

1. In the Radia System Explorer, double-click **NOTIFY**.

2. Right-click the appropriate Notify domain or the **RETRY** domain for which you want to see the status.

3. Click **Status Delete** to delete the status information.

   OR

   Click **Status Display** to display the Status Monitor.
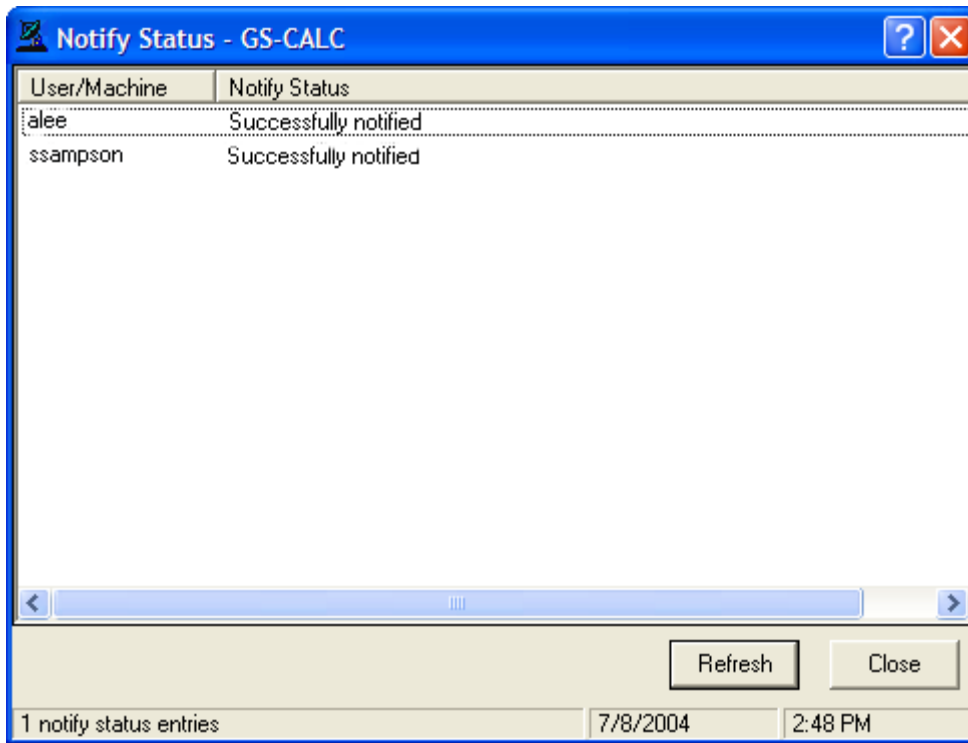
Figure 8.27 ~ Notify Status Monitor.

4. Click **Refresh** if you think the status might have changed.

OR

Click **Close** to close the Status Monitor.

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

Radia can issue a wake-up packet to remotely power-on client computers that have been configured for Wake-On-LAN (WOL).

The Radia System Explorer facilitates the configuring of the WOL client computers to which you want to assign *Drag-and-Drop Notify*-eligibility (DDN).

Note
In order to perform Drag-and-Drop Notify for Wake-On-LAN clients, two settings must be added to the Radia Configuration Server Settings file. See the *Radia Configuration Server Guide* for additional information.

Assigning Clients using DDN

The ease and straight-forwardness of the drag-and-drop functionality dramatically simplifies assigning DDN-eligibility to WOL clients. Use the Radia System Explorer to connect Notify instances to a USER, WRKGRP, DEPT, and SERVICE.

In order to take advantage of the drag-and-drop feature to assign clients for WOL Notify, use the Radia System Explorer. Once you have accessed the Radia System Explorer, you can simply click and hold a Notify instance, drag it to the instance to which you want it connected, and release (drop) it onto it, thereby creating the connection. Refer to the *Radia System Explorer's Guide* for comprehensive information about performing drag-and-drop connections.

Overview of Radia System Explorer Steps

The client computer's MAC (Media Access Control) address and sub-net value must be reported in the ZCONFIG object of the PROFILE file. (These values are LADAPT01 and SUBNET01, respectively, in the ZCONFIG object.)

This section outlines the steps you must take once you have accessed the Radia Database with the Radia System Explorer.

To configure a WOL client for DDN

1. From the **Start menu**, select **Programs, Radia Administrator Workstation, Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

3. The **Radia System Explorer** window opens.

4. Double-click **PRIMARY**.

5. Navigate to the **POLICY** domain.

6. Within the **POLICY** domain, create a new **USER** instance, for example, USER1.

7. Perform a Client Connect to the Radia Configuration Server with the user ID, USER1. This will populate the database with the necessary client information.

> **Note**
>
> In order to perform DDN, the user must exist, with the ZCONFIG and ZMASTER objects, in the Radia Configuration Server's PROFILE file.

Next, you will need to verify the values in the **HARDWARE_SCAN**.

8. Open the **HARDWARE_SCAN** object under **PROFILE.USER1.ZCONFIG**, and check that the **LADAPT01** and **SUBNET01** attributes are present.

Figure 8.28 ~ Sub-net value.

Note		

In order to perform the DDN for WOL to "wake up" this machine you will need to shut down the client machine (USER1).

Next, you will need to create and configure a new instance for the Drag-and-Drop Notify in the Radia Database.

9. Use the Radia System Explorer to create a new instance (for example, DDN_WOL) under PRIMARY.SYSTEM.ZCOMMAND.
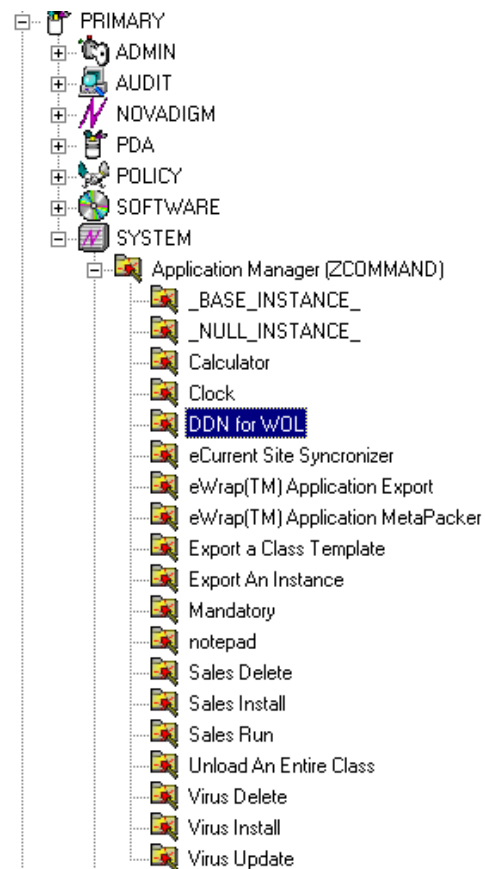
Figure 8.29 ~ ZCOMMAND instance: DDN for WOL.

10. Specify the following parameters:

ZCMDPRMS: `radskman ip=<managerip>, port=<mgr_port>`
                Note: For more information on **radskman**, see *Specifying the Command Line* on page 365.

ZCMDTYPE: `EXE`

ZCMDNAME: `NOTIFY`
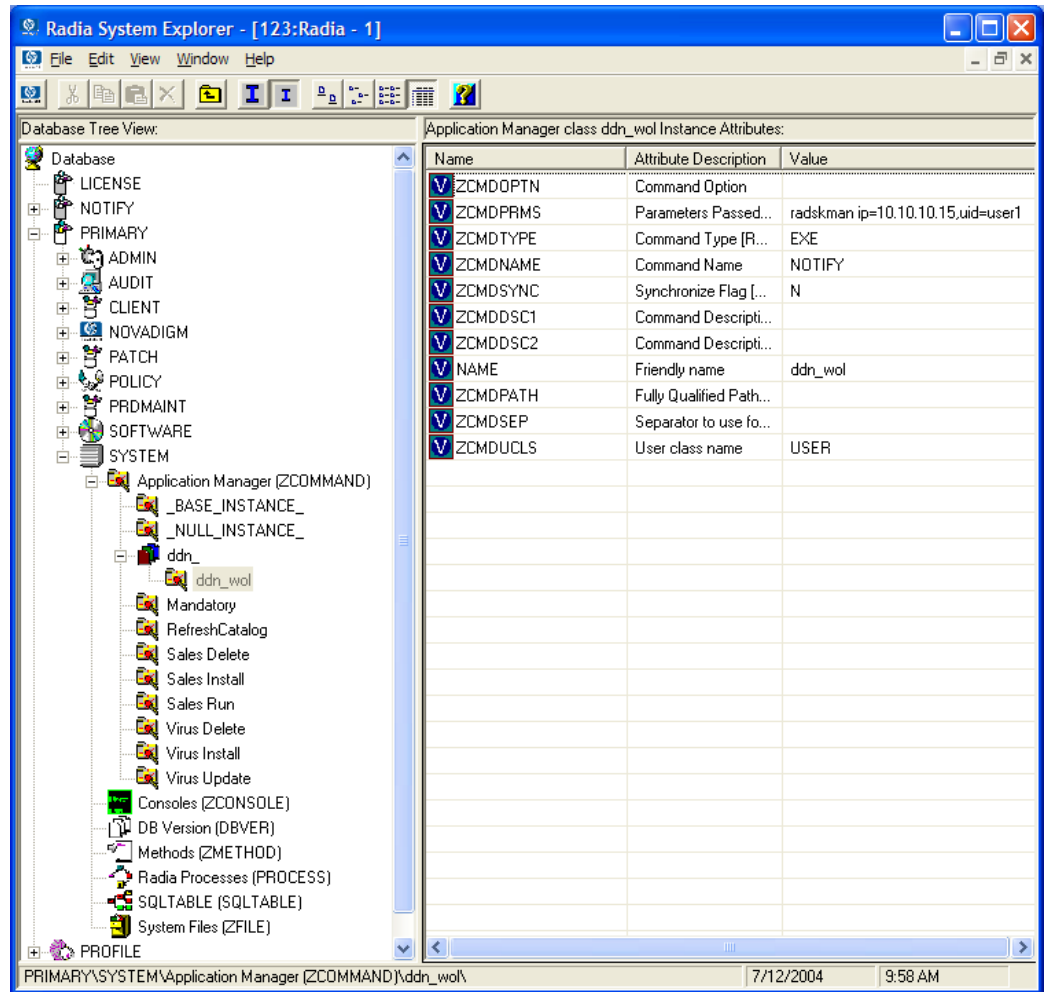
Figure 8.30 ~ View of the DDN for WOL Instance and its attributes.

11. Set up the POLICY.USER1 instance for DDN by dragging the instance to the ZCOMMAND.DDN_WOL instance and dropping it on to it.

You have successfully configured a new WOL user for Drag-and-Drop Notify.

To verify the success of the configuration

On the Client Machine

1. Wait approximately ten minutes before checking the client log.

2. Access **C:\Program Files\Novadigm\Log**, check the RADNTFYD.log and verify the times of the following entries. These should be the same time as when you did the Notification.

```
SyncObjFrmDisk    EDM009461 01.254 13:40:30 Total [0001] pools restored (v161)
EDMNTFYD          EDM000001 01.254 13:40:30 Password verification has not been requested
EDMNTFYD          EDM000001 01.254 13:40:30 Path restricted to EDMSYS subdirectory
EDMNTFYD          EDM000001 01.254 13:40:30 Userid verification has been disabled
Pooltab_replace   EDM009453 01.254 13:40:30 Pool [C:\PROGRAM~1\NOVADIGM\LIB\ZLOCAL.EDM] has
     [5] variables and [1] heaps of size [1024] each (v153)
Edm_ObjPtr_Save   EDM009443 01.254 13:40:30 Object [ZLOCAL   ] being saved (v143)
EDMNTFYD          EDM000001 01.254 13:40:30 UID: user1
EDMNTFYD          EDM000001 01.254 13:40:30 CMD: radskman mame=rad_manager,dname=software,
     ip=208.244.231.61,port=3464,startdir=$machine,ulogon=n,hreboot=y
nvd_exec          EDM000512 01.254 13:40:30 Module Information: Rev 1.105 Oct 30 2000
     11:18:08
nvd_exec          EDM000010 01.254 13:40:30 NVD_EXEC CALLED: Program [C:\PROGRA~1\NOVADIGM
     \radskman]
PARAMETERS[mname=rad_manager,dname=software,ip=208.244.231.61,port=3464,startdir=$MACHINE,
     ulogon=n,hreboot=y]
EDMNTFYD          EDM000001 01.254 13:40:31 Closing socket #44

CloseLogFile      EDM000001 01.254 13:40:31 Closing log file on [Tues Sep 11 13:40:31 2001]
```

Figure 8.31 ~ Excerpt from the RADNTFYD.LOG.

In the Radia Configuration Server

1. From the Radia System Explorer, right click the NOTIFY domain.

2. Select **Refresh**.

3. Note that a new file, NOTIFY has been created.

> **Note**
>
> The Notify file will be created only upon the first notification event.

4. Verify the notification event.

Radia Self Maintenance

Maintenance for the Radia Clients is available from Technical Support. The maintenance will include import decks for the Radia Database. New instances are created in the PRDMAINT class in the PRDMAINT domain. There will be one PRDMAINT instance for each

PRODUCT_PLATFORM_RELEASE combination. These instances will be connected based on the client's platform and current product level. Once you have decided to roll out the maintenance to the client computers, you can add the service to the user's entitlements.

> **Note**
>
> In previous versions of the product, the maintenance functions were in the NOVADIGM domain. If you are using Radia clients below the 4.x version, you will still use the NOVADIGM domain for maintenance activities. See the appropriate *Radia Application Manager* or *Radia Software Manager Guide* for your version of the Radia client.

To minimize the need for separate PRDMAINT bundles based on different operating systems where the actual maintenance is the same, the ZMASTER.ZOSTYPE variables identify the Windows operating system type or family. The valid values for this variable are: "WIN32_NT" (covering Windows NT, 2000, XP, and 2003) and "WIN_9X" (covering Windows 98).

Usage Notes

■ All packages are disabled by default. This is accomplished by setting a ZSTOP expression to "1" to prevent deployment. Either remove this value for general deployment, or use this ZSTOP expression to restrict its deployment to certain groups.

■ The first REQUIRES connection is reserved for any possible hot fix, a fix sent to you directly by Technical Support is not yet available in a fix or service pack. This package, _HOTFIX, will be used to chain any required fixes (and/or enhancements) and will be maintained by the customer. The second connection is for any locally customized code to be included as part of maintenance.

■ Use the ACTMAINT attribute in the SETTINGS class of the CLIENT domain to specify how you want maintenance processed. You can choose to immediately download and install maintenance (I), download only and install later (D), or prompt users to install maintenance at another time (D). Maintenance *only* runs when the mnt parameter of radskman is set to Y. See *ACTMAINT* on page 284 and the *mnt* parameter on page 351 for more information.

We will provide an updated PRDMAINT instance with each new maintenance pack. The customer is not required to apply all maintenance.

To deploy client maintenance packages

1. A maintenance package is made available on the HP OpenView web site in the form of an export deck.

2. Download the files. There should be at least an xpi and xpr file.

3. Stop the Radia Configuration Server service and copy the export files to the Radia Configuration Server's bin directory.

4. Import the files using the ZEDMAMS utility. For detailed information on the use of this utility, see the *Radia Configuration Server Guide*.

   For example, if you were given two files, MAINT_RAM_40_RC3.XPI and MAINT_RAM_40_RC3.XPR. You might use the following two command lines.

   ```
   ZEDMAMS VERB=IMPORT_INSTANCE,FILE= MAINT_RAM_40_RC3.XPI,PREVIEW=NO
   ZEDMAMS VERB=IMPORT_RESOURCE,FILE= MAINT_RAM_40_RC3.XPR,PREVIEW=NO
   ```

   > **Note**
   >
   > Your command line may vary depending on a number of factors. For detailed information on the use of this utility, see the *Radia Configuration Server Guide.*

5. Restart the Radia Configuration Server.

6. Assign the Maintenance Service to the appropriate users in the POLICY domain.

   > **Note**
   >
   > To run the maintenance portion of a Client Connect process, the mnt parameter of the radskman command line, must be set to Y.

During catalog processing, the client will first process all services found in the PRDMAINT domain, perform arbitration to determine appropriate maintenance, and deploy the maintenance to the maintenance staging directory. The default location for this is c:\Program Files\Novadigm_Maint_.

Radia System Tray

The Radia System Tray icon provides status and statistics information, as well as pause and cancel mechanisms to the subscriber. The Radia System Tray icon sits in listen mode, and accepts requests for the display of dialog boxes and status information that will be shown when the Radia Client needs user interaction. Radia 4.0 now supports French and German for the Radia System Tray.

Note
The Radia System Tray is not supported for Windows 95.

An icon shows in the Radia System Tray area of the Task Bar. By moving your cursor over the icon, you can see one of three states depending on what the Radia Client is doing at the time.

- **Idle**
  When the Radia System Tray is in listen mode, the icon is static.

Figure 8.32 ~ Radia is idle.

- **Active**
  The icon becomes active when the Radia Client is working or when user intervention is required. The icon animates and an informational bubble will appear when you move your cursor over the icon. The bubble provides information on the type of activity that is occurring. If a critical notify occurs, the bubble will popup automatically:
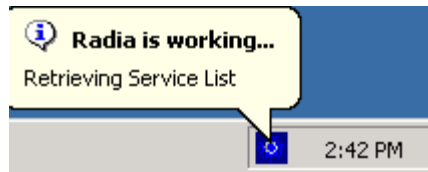
Figure 8.33 ~ Radia is working.

■ **Console View**
The Console View can be launched by the shortcut menu that is available when you right-click on the icon, or by double-clicking on the icon. The Console view appears as shown in the figure below.
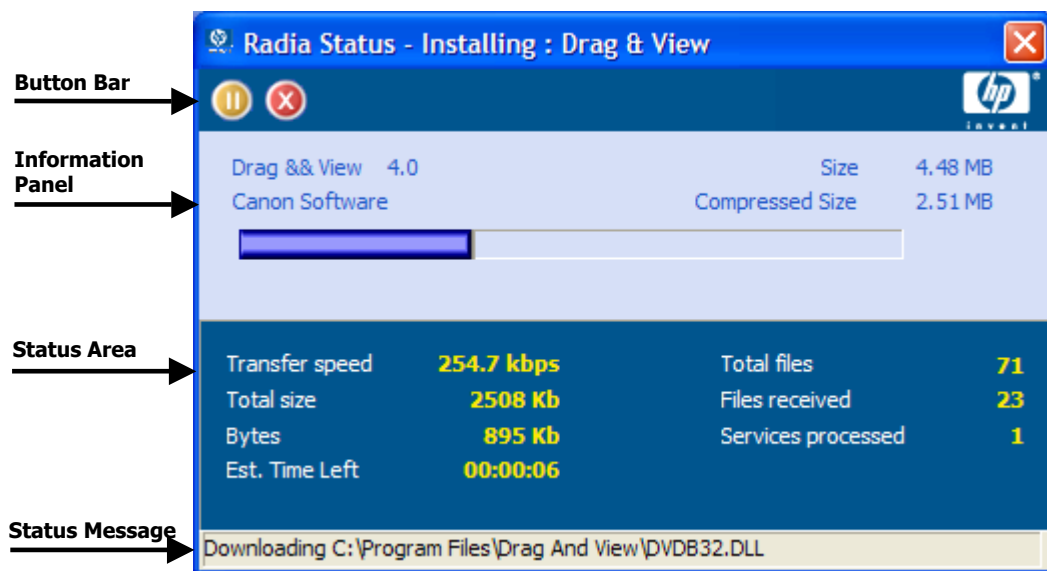
Figure 8.34 ~ Double-click the icon to view the Radia System Tray Console.

The Console View contains the following parts:

■ **Button Bar**
Contains buttons for Pause and Cancel, and a logo that animates when Radia is actively working.

■ **Information Panel**
Contains information about the service that is currently being processed, as well as a progress bar that shows the percentage finished.

■ **Status Area**
Contains statistics about the current processes, including transfer speed, total size of transmission, bytes received, estimated time left of transmission, total files to be transmitted, number of files received, and number of services processed.

■ **Status Message Area**
The Status Message Area shows a message about the current process.

■ **Bandwidth Control**
If you set bandwidth throttling for the Service on the Radia Configuration Server, and you click the bandwidth toggle button in the System Tray Console, a slider for bandwidth control appears. Adjusting the slider will result in the bandwidth throttling value being changed.

Note
The bandwidth control shows when bandwidth throttling is available (based on the throttling type for the service, Adaptive or Reserved). In addition, the bandwidth slider will be displayed if the throttling type is valid and the UIOPTION attribute of the Application (ZSERVICE) instance is set to FULL. FULL is the default value. Set UIOPTION to INFO to show what is happening on the client computer, but disable all the controls so that the subscriber cannot make any changes. Set the UIOPTION to NONE so that no dialog boxes are displayed. Set the UIOPTION using the Radia System Explorer.
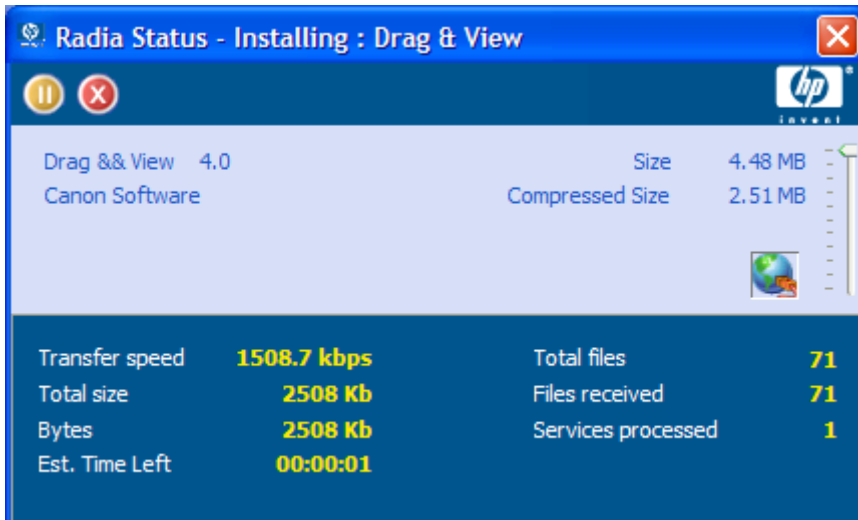
Figure 8.35 ~ Control the bandwidth from the System Tray Console.

Application Alert Messages and Deferrals

Use the Radia System Explorer to show the subscriber that an application has a high priority or to display an additional message. An Application (ZSERVICE) instance can be set to have a high or normal priority. An exclamation point (!) denotes that an application is high priority.

If you are managing a high priority service using the Radia Software Manager with the Radia System Tray, then the alert message will display in the status bubble of the Radia System Tray icon, and the bubble will pop up whether or not your cursor is over the icon.
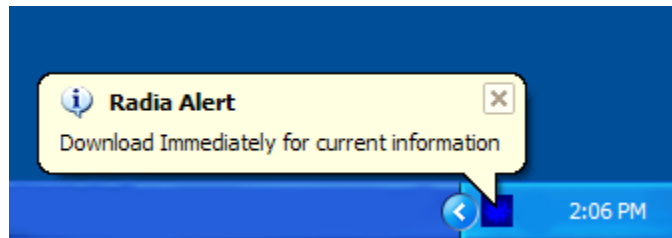
Figure 8.36 ~ High priority status bubble pops in Radia System Tray.

When an application is deployed, an administrator can also display a deferral message based on the network threshold, the data download size, a specified date setting, or a deferral count. When an application has data that needs to be downloaded to the client computer, the Radia Software Manager client will check if the application is configured for deferral. If it is, the Radia Client will check the current bandwidth setting against the administrator specified bandwidth threshold setting. If the current network speed *is less* than the Network Threshold (DT) value, a deferral message will be displayed asking the subscriber if he wants to defer the deployment. The administrator can configure the number of times an application can be deferred, the date an application can be deferred until, or a minimum byte count to alert on. If the number of deferrals or the deferral date has been reached, the application will be installed or updated without displaying a deferral message. If the size of the data is less than the minimum byte count, the alert panel will be skipped.

If the application has been configured for a deferral, and all of the requirements listed below are met, the Radia Client will display the deferral dialog box.

- The Alert Mode (DM) is configured for the current operation: Install, Update, or Both.
- The current network speed is lower than Network Threshold Speed (DT).
- The UIOPTION attribute in the ZSERVICE instance is *not* set to NONE.
- The data to be downloaded is greater than the administrator specified minimum byte count, (DBT) and lower than the specified maximum byte count (DAT).
- If specified, the deferral date, Allow Install Deferral up to (DI), or Allow Update Deferral up to (DU) has been reached.

OR

■   The number of deferrals allowed (DN) has been reached.

If these requirements are met, and you are using the Radia Software Manager, the following will be displayed:
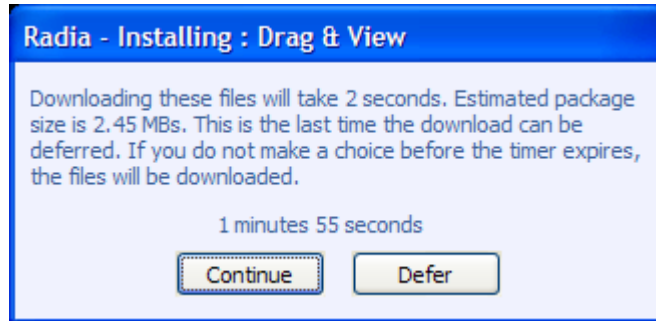
Figure 8.37 ~ Radia Software Manager Defer alert display.

The subscriber can choose to defer the action or to continue with it.

> **Note**
>
> If the timeout value is exceeded, the action will be taken that is identified in the DA (Action on Timeout Cont/Defer) attribute.

To implement an Application Alert or Deferral, you will need to create an instance in the Alert/Defer (ALERTDEF) class, and connect that instance to the appropriate Application (ZSERVICE) instance.

Creating an Alert Message or Deferral Instance

The Alert/Defer (ALERTDEF) class has been added to the SOFTWARE domain in the Radia Database to configure application alerts. In order to configure an alert, you will need to create an instance in the Alert/Defer (ALERTDEF) class.

To create an instance of the Alert/Defer (ALERTDEF) class

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

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**. The **Radia System Explorer** window opens.

3. Double-click **PRIMARY**.

4. Double-click **SOFTWARE**.

5. Right-click **Alert/Defer (ALERTDEF)**. A shortcut menu opens.

6. Click **New Instance**. The **Create Instance** dialog box opens.
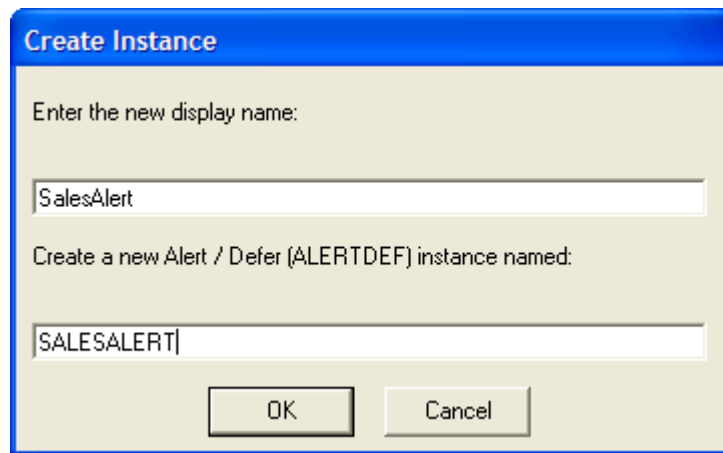
Figure 8.38 ~ Create a new instance.

7. Type in a name for the new instance. In our example, we create an instance called **SalesAlert**.

8. Click **OK**.

The new instance is created.

Configuring an Alert Message or Deferral

Once the instance is created, you need to configure it for your Alert. The Alert/Deferral (ALERTDEF) class includes two sample instances, Dial Up Sample Defer, and LAN Sample Defer.

To configure an Alert/Deferral (ALERTDEF) instance

1. Use the Radia System Explorer to navigate to the Alert/Defer (ALERTDEF) instance you want to edit.

2. Double-click the instance. In this example, we are editing the **SalesAlert** instance.
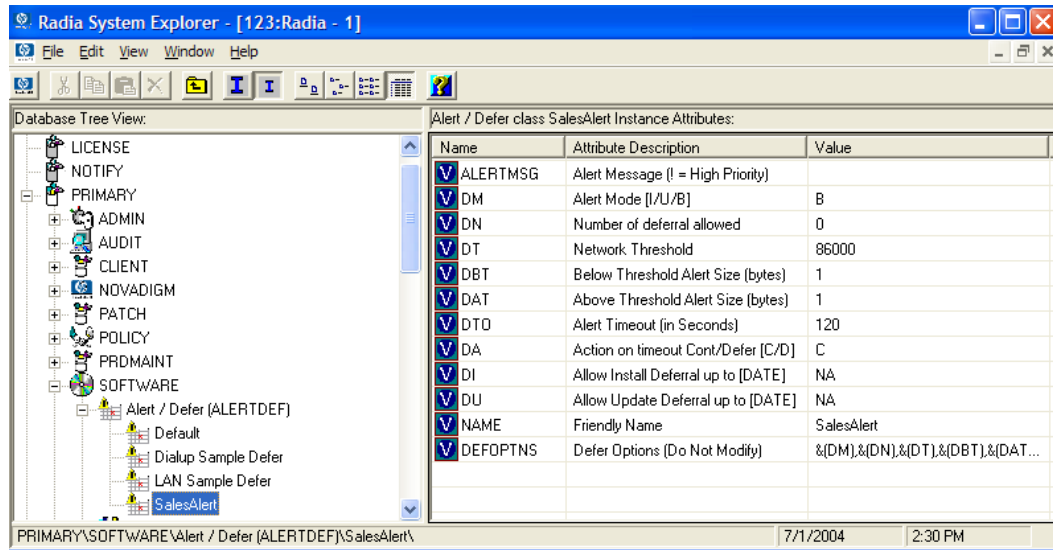
Figure 8.39 ~ Double-click the instance you want to edit.

3. Double-click the variable you want to edit. See *Table 8.11 ~ Variables in the ALERTDEF class* below for information on the attributes for this class.

Table 8.11 ~ Variables in the ALERTDEF class		
Variable	**Description**	**Default Value**
ALERTMSG	**Alert Message** An exclamation point (!) preceding "Service Alert Message" denotes a high priority message.	

Table 8.11 ~ Variables in the ALERTDEF class

Variable	Description	Default Value
DM	**Alert Mode [I/U/B]** Set to **I** for Install. Set to **U** for Update. Set to **B** for either Install or Update.	B
DN	**Number of deferrals allowed** Maximum number of deferrals allowed before the action specified in DA (Deferral Action) is taken.	0
DT	**Network Threshold** Network bandwidth threshold in bytes. The current network speed must be less than this value to meet the deferral requirement.	86000
DBT	**Below Threshold Alert Size (bytes)** The size of the file to be downloaded must be greater than this value to meet the deferral requirement.	50000
DAT	**Above Threshold Alert Size (bytes)** The size of the file to be downloaded must be less than this value to meet the deferral requirement. If the value is 0, this attribute is ignored.	0
DTO	**Alert Timeout (in seconds)** Specify in seconds the amount of time to display the Defer Alert dialog box. After the timeout is reached, the action specified in the DA (Action on timeout) variable, will be taken.	120
DA	**Action on timeout Cont/Defer [C/D]** Specify **C** to continue with the specified action if the subscriber does not respond to the defer alert dialog box by the value specified in the **DTO** (Alert Timeout). Specify D to defer the specified action.	C
DI	**Allow Install Deferral up to [DATE]** After this date is reached, the option to defer installation will no longer be available. The application will be installed. This should be in the format YYYYMMDD.	NA
DU	**Allow Update Deferral up to [DATE]** After this date is reached, the option to defer an application update will no longer be available. The application will be updated. This should be in the format YYYYMMDD.	NA
Name	**Friendly Name** Friendly Name for the instance.	Default
DEFOPTNS	**Defer Options (Do not Modify)** This attribute is used to resolve the values of the other attributes of this class. *Do not modify*.	&(DM),&(DN),&(DT),&(DBT),&(DAT),&(DTO),&(DA),&(DI),&(DU)

In this example, we want to add an alert message with high priority. To do this, double-click the **ALERTMSG** variable in the list view.
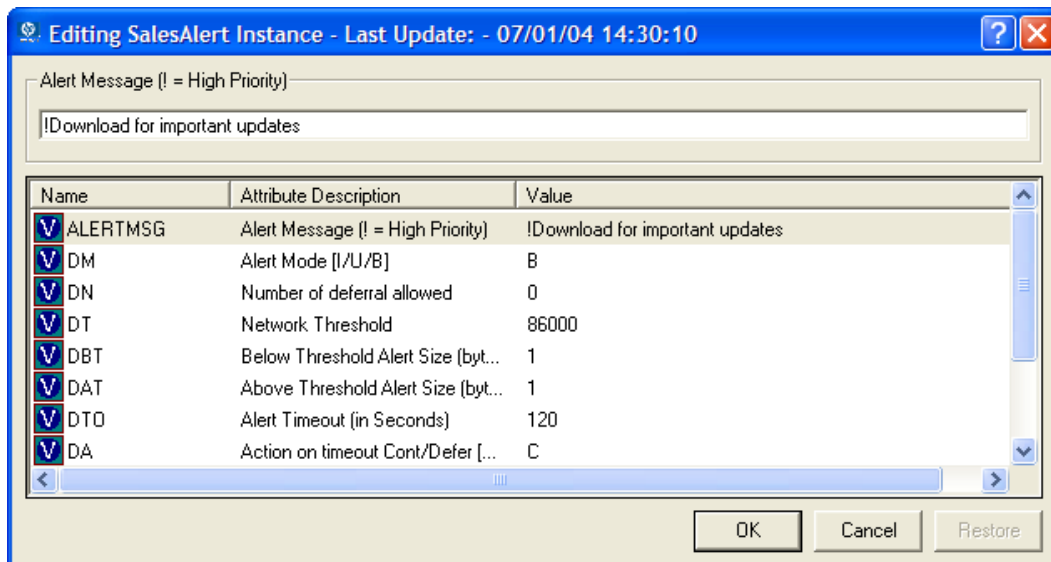
Figure 8.40 ~ Edit the SalesAlert instance.

4. Type the message you want displayed.

5. Click on the next attribute, and type in the appropriate value.

6. Click **OK** when you are finished editing the attributes. The **Instance Edit Confirmation** dialog box opens.

7. Click **Yes** to confirm the changes.

The changes are made to the Alert/Defer (ALERTDEF) instance.

Once the Alert/Defer (ALERTDEF) instance is created, you need to connect the Alert/Defer (ALERTDEF) instance to an Application (ZSERVICE) instance. To do this, use the Radia System Explorer to click and drag the Alert/Defer (ALERTDEF) instance to the appropriate Application (ZSERVICE) instance. For additional information on using the Radia System Explorer, see the *Radia System Explorer Guide*.

Local AIP support for the Radia MSI Redirector

The Radia MSI Redirector is a specialized, localhost HTTP server that accepts and satisfies file requests made during an MSI installation. Upon receiving the HTTP request from MSI, the Redirector retrieves the file from its local cache (if it exists) or requests the file from an upstream Radia Configuration Server or Radia Proxy Server. This process requires that the MSI installation support HTTP, and some vendors, including Microsoft, have removed HTTP support from their products' installations. Without HTTP support, the MSI install would not be able to directly request files from the Redirector. In order to continue to leverage the benefits of the Redirector, another module has been developed for the Redirector. This module adds another level of redirection at the local file system level. When using this method, MSI is told that the AIP is local, and requests files directly from the file system. This request is captured and forwarded to the MSI Redirector, which satisfies the request in its normal manner. The file is then placed in the defined 'local' AIP, where MSI can process it. This local AIP is temporary, and is removed after the installation is completed.

 This additional functionality has been added to the MSI Redirector as part of the Radia 4.0 Release. This provides the following advantages:

■   Increased level of flexibility and performance.

■   Ability to apply date/time stamps and support long file names.

■   Ability to run Custom Actions that do not support http as an Install Point.

To enable using the Local Application Installation Point

1. Use Radia System Explorer to navigate to the MSI Resources (MSI) class in the SOFTWARE domain. Each MSI application will have an msi instance and an idx instance.

2. Right-click the MSI instance and select **Edit Instance**.

3. To turn on this feature, set **MSIDRIVR** (Use Local AIP [Y/N]) to **Y**.

Note
If MSIDRIVR is not present in your database, create it in the MSI Resources (MSI) class as a 1-byte variable with a description of "Use Local AIP [Y/N]". We recommend backing up your database before making changes to a class template. For information on editing class templates, see the *Radia System Explorer Guide*.
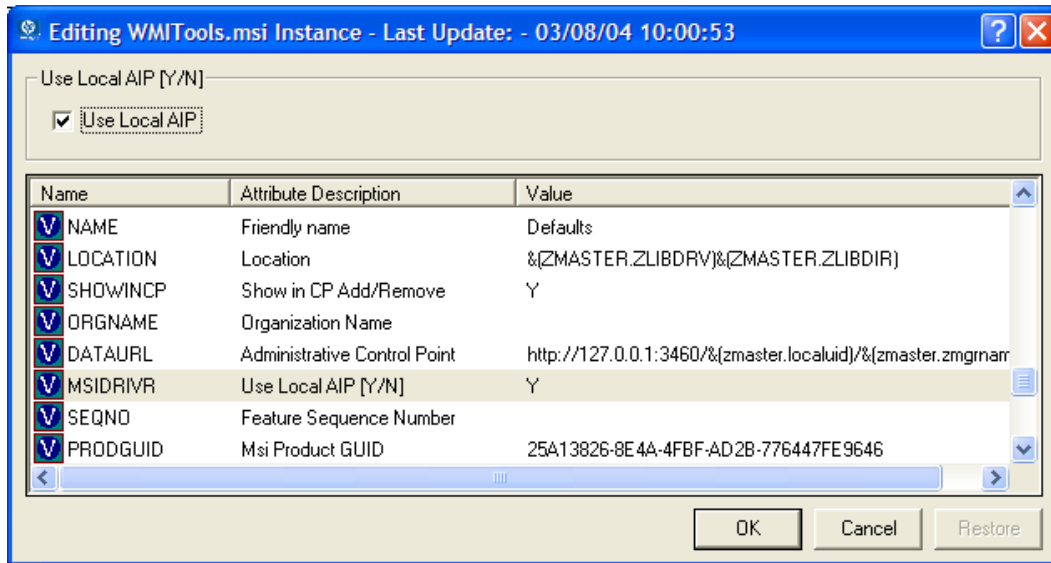
Figure 8.41 ~ Set MSIDRIVR to Y.

4. Click **OK**.

5. Click **Yes** to confirm the changes.

6. Configure the SETTINGS.LOCALAIP variable in COP to control the destination of the local AIP folder on the client desktop. For example, c:\localaip.

Caution
We recommend keeping your LOCALAIP as short as possible to accommodate AIPs that have deep directory structures.

Internet Proxy Detection

Radia has the ability to detect an Internet proxy when an Internet proxy is used. The Internet proxy's address is then stored in PROXYINF.EDM located in the client computer's IDMLIB directory. The default location of IDMLIB is **<System Drive>:\Program Files\Novadigm \Lib**. The next time the client computer connects to the Radia Configuration Server the specified Internet proxy will be used. To use this feature, you must enable your Radia Client to use and discover an Internet proxy. If you are using the Radia Application Manager, set USEPROXY to **Y**, and DISCOVER to **Y**, in the PROXYINF.EDM client object.

About Radia Proxy Servers

Use Radia Proxy Servers to load a portion of the work required to deploy applications from the Radia Configuration Server to another server computer. You may want to do this for the following reasons:

■ The Radia Proxy Server may be closer to the clients on the network.

■ You may want to reduce the load on the Radia Configuration Server.

When using Radia Proxy Servers, the software to be distributed is copied to the Radia Proxy Server. The Radia Proxy Server then provides the software to those Radia Clients that are not required to obtain their software from the Radia Configuration Server. The potential benefit of Radia Proxy Servers must be evaluated individually for each server and its Radia subscriber computers.

For more information, refer to the *Radia Proxy Server Guide*.

Summary

■ Carefully plan and test your application deployment strategy to determine the best distribution method for your subscribers.

■ Use the timer to deploy an application at a specific time or interval.

■ Use Notify to update or remove an application that has already been installed using Radia, or to notify users of an update via e-mail.

■ Use a Drag-and-Drop Notify command to initiate a Notify to one or more subscribers immediately. The benefit of this type of Notify is that the application does not have to be installed on the client computer to perform the Notify. You might use this type of notify to install software, update all mandatory services, or even run an executable on the client computer.

■ Use the Version Group Editor to manage versioned applications.

■ Radia supports Windows Installer-enabled Applications.

■ Use the Feature Set Editor to control customizations of MSI packages.

■ Test deployments in a safe environment before distributing them to your subscribers.

■ Consider using Radia Proxy Servers to minimize network traffic or to minimize work on the Radia Configuration Server.

9

Radia Client Directories and Objects

At the end of this chapter, you will:

■ Know the directory structure of the Radia Client.

■ Be familiar with core Radia Client objects.

■ Know where client objects are stored.

Although you will tailor this strategy to meet your organization's needs, we recommend that you review this guide for a comprehensive understanding of the Radia Software Manager. This chapter covers Radia Client objects.
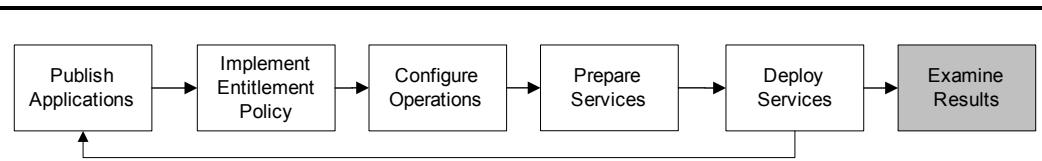
Figure 9.1 ~ Tasks completed in this guide.

Radia Client Directory Structure

Below is an example of the directory structure on a Radia Software Manager client computer following the installation of the Radia Software Manager, a forced system (machine) connection from the client to the Radia Configuration Server, and the installation of a mandatory application.
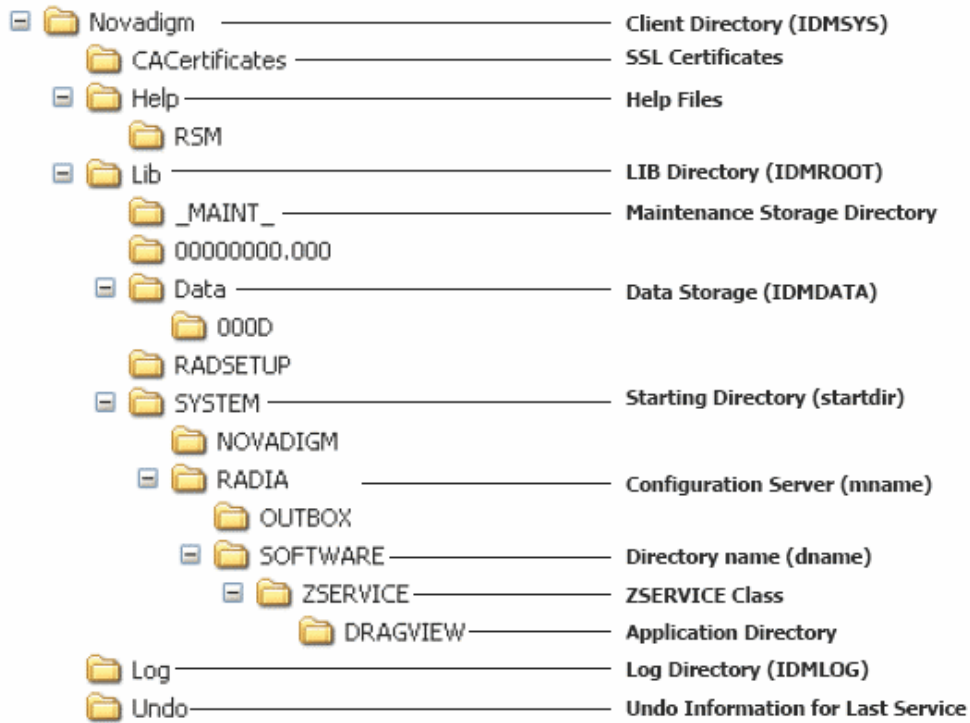
Figure 9.2 ~ View the client directory tree.

Note
The term *computer* is used to refer to a workstation or server.

Radia Software Manager Directories

The initialization settings for the Radia Software Manager for Windows are located in the [NOVAEDM] section in the NVD.INI file, on the client computer. NVD.INI is located, by default, in the IDMLIB directory and this location is stored in the registry key HKLM\Software\Novadigm\Radia\Settings\Inipath.

Table 9.1 ~ NOVAEDM Parameters	
Parameter	**Description**
IDMDATA	When Radia installs software, the Radia Client temporarily stores compressed files received from the Radia Configuration Server in this folder.
	Once the files are decompressed and installed on the client computer, the compressed files are erased.
	Default: *SystemDrive*:\Program Files\Novadigm\Lib\Data\
IDMLIB	Dynamic directory that stores the objects for the service currently being managed.
	Default: *SystemDrive*:\Program Files\Novadigm\Lib\
IDMSYS	Stores the Radia Client executables, such as .EXE and .DLL files.
	Default: *SystemDrive*:\Program Files\Novadigm\
IDMROOT	The base directory for IDMLIB. This is a static path.
	Default: *SystemDrive*:\Program Files\Novadigm\Lib\
IDMLOG	Stores the Radia Client logs.
	Default: *SystemDrive*:\Program Files\Novadigm\Log\

About Radia Client Objects

When a client computer connects to the Radia Configuration Server, information is exchanged between the client and the Radia Configuration Server. This exchange is called *resolution*. During resolution, Radia checks the status of services, and updates the Radia Configuration Server with information from objects stored on the client computer.

Radia Client Objects are stored in the IDMLIB directory on the client computer. After installing the Radia Software Manager client and connecting to the Radia Configuration Server, you can use Radia Client objects to answer questions such as:

❑ What is the hardware configuration of the client computer?

❑ Was the service successfully installed?

❑ When was the service installed?

❑ What is the client computer's name, and who was the last user logged on?

❑ What are the possible data sources for this client computer?

While there are multiple Radia objects on a client computer at any time, there is a core group of objects that supply information about and the status of the current Client Connect. *Table 9.2 ~ Core Client Objects* below lists these core objects. The table includes information on when the object is created or updated, and a brief summary of what the object includes. Each object listed has its own section in this chapter including a table listing its attributes. There are other objects created during the Client Connect, but only the most commonly used ones are noted here. Check the HP OpenView web site for information on other client objects.

Table 9.2 ~ Core Client Objects		
Object	**When created or updated**	**Type of Information included**
ZCONFIG	ZCONFIG is created at start of Client Connect process. See *Table 9.3 ~ ZCONFIG Attributes* for more information.	Contains basic hardware information for the client computer such as processor, operating system, and drives.
SYNOPSIS	This object is transferred to the Radia Configuration Server at the end of the Client Connect. **Note**: Client Operations Profiles must be enabled for this object to be present. See *Table 9.4 ~ SYNOPSIS Attributes* for more information.	RADSKMAN stores a job summary in the SYNOPSIS object. It reports some of the parameters from the RADSKMAN command line and information on the number of files and bytes added, removed, and repaired.
SAPSTATS	Updated by any network bound modules that need to access the Server Access Profile (SAP) such as RADCONCT, RADSTGRQ, and RADSTGMS. RADSKMAN deletes the SAPSTATS object at the beginning of the job. **Note**: Client Operations Profiles must be enabled for this object to be present. See *Table 9.5 ~ SAPSTATS Object Attributes* for more information.	The SAPSTATS object has one instance for each of the client computer's Server Access Profiles (SAP). It summarizes information for each SAP such as speed, number of files sent and received, and the role of the SAP.
PREFACE	PREFACE is sent to the Radia Configuration Server at every phase of a **radskman** process including: • Client operations profile resolution • Client Self Maintenance resolution • Catalog resolution • Single service resolution (This can happen multiple times depending on what services are processed.) • Outbox flush See *Table 9.6 ~ PREFACE Object Attributes* for more information.	Contains core information about each invocation of **radskman**.

Table 9.2 ~ Core Client Objects		
Object	**When created or updated**	**Type of Information included**
SMINFO	SMINFO is created during Client Operations Profiles resolution, but does not require Client Operations Profiles. See *Table 9.7 ~ SMINFO Attributes* on page 430 for more information.	SMINFO collects information that is independent of the hardware and software installed on the computer, and some network information.

Radia Client Version

Some of the objects described in this book apply only to Radia Clients version 3.1 and above. To verify the client's versions, open the connect.log file in the IDMLOG directory on any operating system using a text editor and search for "version". On Windows Operating systems, you can also check the Version tab of the Properties of **radskman** in the IDMSYS directory.

Using the Radia Client Explorer to View Objects

Radia Client Explorer is installed as a component of the Radia Administrator Workstation. Use it to view objects in the IDMLIB directory. You can view any object if you have access to the client computer's IDMLIB directory. Otherwise, you may need to manually retrieve the object file, and store it on your Radia administrator computer.

To view an object using the Radia Client Explorer

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia Client Explorer**.

The **Radia Client Explorer** opens.
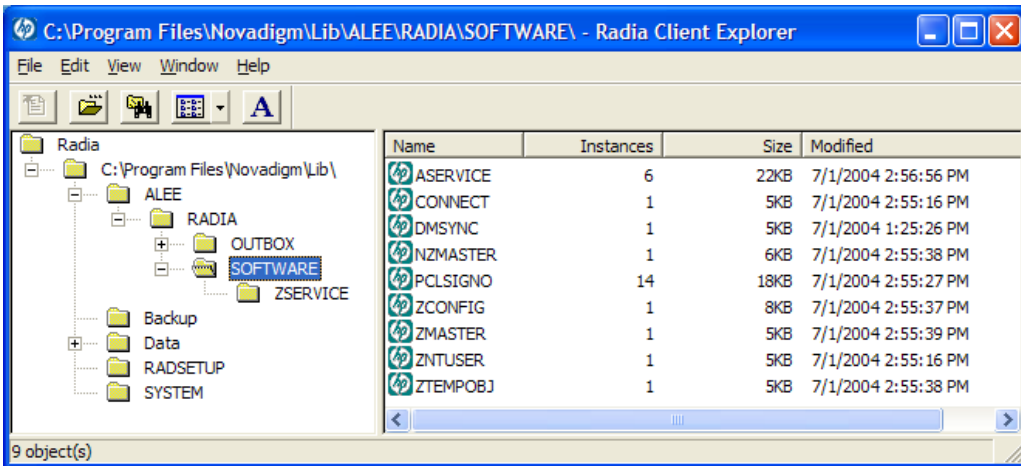
Figure 9.3 ~ Radia Client Explorer expanded to SOFTWARE.

2. If necessary, from the **File** menu, select **Change Directory** to navigate to the client computer's IDMLIB directory or to the directory where the object is stored.

3. Double-click the object's name in the list view.

   The Radia Client Explorer displays the selected object.

4. Click **Save/Exit** to close the dialog box.

Hardware Configuration Information (ZCONFIG)

The ZCONFIG object stores hardware configuration information from the client computer. Use the Radia Client Explorer to view the ZCONFIG object. The following table describes the attributes of ZCONFIG arranged in alphabetical order. These attributes may vary depending on the configuration of the client computer.

Note

The ZCONFIG object is sent to the Radia Configuration Server automatically for viewing with the Radia System Explorer. If you do not want this object sent to the Radia Configuration Server, set the POLICY.USER.ZCONFIG attribute to N in the base instance. This will stop collection from ALL users. The object will still exist on the client computer.

Table 9.3 ~ ZCONFIG Attributes

Attribute	Description	Example
BOOTDRV	Designation of the boot drive	C:
BOOTDRVI	Type of boot drive	IDE
DHCPSR0n	IP address of the DHCP Server of LADAPT0n adapter	10_10_10_6
DHCPSVR	IP address of the DHCP Server for the current LAN adapter	
DNSDMN01	Domain name used by the client computer for LADAPT0n adapter	
DNSDOMN	Domain name currently being used	
DNSHNM01	Host name used by the client computer for LADAPT0n adapter	
DNSHOSTN	Hostname currently being used	
GATEWY01	Gateway Address of network adapter 1	10.1.1.1
HALCOMP	Company of HAL.DLL	Microsoft Corporation
HALDATE	Date and time of HAL.DLL	20010817 19:00:00
HALFNAME	Original name of HAL.DLL	Halacpi.dll
HALFVER	Internal version of HAL.DLL	5.1.2600.0 (xpclient.010817-1148)
HALINAME	Name of HAL.DLL	Halacpi.dll
HALLANG	Language of HAL.DLL	English (United States)
HALPNAME	Product name of HAL.DLL	Microsoft® Windows® Operating System
HALPVER	Product version of HAL.DLL	5.1.2600.0
HALSIZE	Size of HAL.DLL	78464
IPADDR01	IP address of network adapter 1	10.1.1.32
LADAPT01	LAN Adapter 1	00C04F6D76B4
REBOOTD	Reboot Date	20010126
REBOOTT	Reboot Time	13:12:30
SCANTYPE	Type of hardware scan	WMI
SUBNET01	Subnet Mask for LADAPT01	255.255.255.0
ZGATEWAY	Gateway Address	&(GATEWY01)
ZHDWBIOS	BIOS type	07/27/98 Phoenix ROM BIOS PLUS Version 1.10 A02
ZHDWCDDR	Drive letter of CD-ROM drive	H:
ZHDWCOMP	Computer Name	ALEE
ZHDWCPU	CPU type	GenuineIntel
ZHDWCPUN	Number of CPUs installed	1
ZHDWCPUS	Current CPU speed	350MHz

Table 9.3 ~ ZCONFIG Attributes

Attribute	Description	Example
ZHDWCTYP	Computer Type (Desktop or Laptop)	LAPTOP
ZHDWD00	Drive name for drive 00	C:
ZHDWD00C	Drive classification for drive 00	Fixed
ZHDWD00F	Current free space on drive 00	321,093,632
ZHDWD00S	File system on drive 00	FAT
ZHDWD00T	Total space for drive 00	2,146,467,840
ZHDWD01	Drive name for drive 01	D:
ZHDWD01C	Drive classification for drive 01	Fixed
ZHDWD01F	Current free space on drive 01	1,116,372,992
ZHDWD01S	File system on drive 01	FAT
ZHDWD01T	Total space for drive 01	2,146,467,840
ZHDWDF_A	Information for floppy drive A	3.5, 1.44MB, 512 bytes/sector
ZHDWDLST	List of assigned drive letters	A, C, D
ZHDWDNUM	Number of drive letters assigned	3
ZHDWFPU	Current FPU type	x86 Family 6 Model 5 Stepping 2
ZHDWIPAD	IP Address	&(IPADDR01)
ZHDWKYBD	Keyboard type	IBM enhanced (101- or 102-key); Number of function keys = 12
ZHDWLANA	LAN Adapter	&(LADAPT01)
ZHDWLANG	Language	English (United States)
ZHDWMEM	Total physical memory (RAM)	133,619,712
ZHDWMEMF	Current total free memory (RAM)	66,543,616
ZHDWMOUS	Mouse (pointing device) type	msps2
ZHDWNET1	Network adapter 1 information	3Com Fast EtherLink XL 10/100Mb TX Ethernet NIC (3C905B-TX)
ZHDWNNET	Number of network adapters installed	1
ZHDWOS	Operating system and version	Windows_NT Version 4.0 [Build 1381]
ZHDWOSCL	Operation system classification (Workstation or Server)	Workstation
ZHDWOSDB	Operating system build	1381
ZHDWOSOG	Operating system organization	ASD Foods, Inc.
ZHDWOSOW	Operating system owner	Anna Lee
ZHDWPA00	Printer 00 information	\\ASD\ACCOUNTING on Port: \\ASD\ACCOUNTING Driver: HP LaserJet 4 Plus

Table 9.3 ~ ZCONFIG Attributes		
Attribute	**Description**	**Example**
ZHDWPA01	Printer 01 information	Apple LaserWriter Pro 810 on Port: LPT1: Driver: Apple LaserWriter Pro 810
ZHDWPPAR	Number of parallel ports	1
ZHDWPPRN	Number of printers available	2
ZHDWPSER	Number of serial ports	2
ZHDWSVCP	Service pack applied	Service Pack 4
ZHDWVIDO	Video type	ATI Technologies Inc. 3D RAGE PRO
ZHDWXPAG	Page size	4096 bytes
ZHWCPU01	First CPU type	GenuineIntel
ZHWFPU01	First FPU type	x86 Family 6 Model 5 Stepping 2
ZHDWVIE	Microsoft Internet Explorer Version	6.0.2600.0000
ZHDWVMSI	MSI Version	2.0.2600.0
ZHDWVRES	Video Resolution	800x600
ZMODEM	Modem present?	Y
ZOBJRRC	Resolution return code	000
ZOBJRSTY	Resolution type	C
ZUSERID	User ID or computer name	ALEE

Client Operations Profile Summary (SYNOPSIS)

The SYNOPSIS object is created on clients using Client Operations Profiles. The SYNOPSIS object summarizes the most recent Client Connect. Use the SYNOPSIS object to confirm the success or failures of the Client Connect process. See *Chapter 6: Configuring Client Operations Profiles* starting on page 261 for more information on implementing Client Operations Profiles.

Table 9.4 ~ SYNOPSIS Attributes	
Attribute	**Description**
STARTIME	Start time in ISO8601 time format e.g 1997-08-15T11:12:00-0400
ENDTIME	End time in ISO8601 time format
EXITCODE	Exit code from the job
ERRORMSG	Text message corresponding to the EXITCODE described in *Radia Management Applications Messages and Codes Guide*
PRIORAPP	Total number of applications that existed in the service list (installed/not installed) before this job started

Table 9.4 ~ SYNOPSIS Attributes

Attribute	Description
PRIORINS	Total number of installed applications that existed in the service list before this job was started
PRIORERR	Total number of applications in the service list that have errors before this job started
CURRAPP	Number of applications in the service list after the job completed
CURRINS	Number of applications in the service list that have been installed
UPDNUM	Number of updates found in the service list
UPDSKIP	Number of updates skipped
UPDDONE	Number of updates processed
UPDFAIL	Number of updated that failed
ADDNUM	Number of new applications found in the service list
ADDSKIP	Number of installs skipped (possibly optional applications)
ADDDONE	Number of installs processed
ADDFAIL	Number of installs that failed
DELNUM	Number of deletes found in the service list
DELSKIP	Number of deletes skipped
DELDONE	Number of deletes processed
DELFAIL	Number of deletes that failed
VERNUM	Number of applications that were verified
VERSKIP	Number of verifications skipped
VERDONE	Number of verifications processed
VERFAIL	Number of verifications that failed
REPNUM	Number of applications that were repaired
REPSKIP	Number of repairs skipped
REPDONE	Number of repairs processed
REPFAIL	Number of repairs that failed
CREFRESH	Catalog Refreshed (Y/N)
JOBID	Jobid passed in on the command line via notify
ZUSERID	Userid for this job
ZCONTEXT	Context of this job (M – Machine or U – User)
MACHNAME	Machine name of the client computer from where this was run
USEREXEC	User that executed the job
CMDLINE	Command line parameters used to execute this job

Service Access Profile Status (SAPSTATS)

The SAPSTATS object is generated on clients using Client Operation Profiles, and is used to report the Server Access Profile (SAP) status and usage statistics from the Radia Client. The SAPSTATS object contains all the variables defined in the SAP class in the Radia Configuration Server database along with the following usage related variables. For more information on the SAP class, see *Chapter 6: Configuring Client Operations Profiles* starting on page 261.

Table 9.5 ~ SAPSTATS Object Attributes

Attribute	Description
BANDWDTH	Percentage of bandwidth to use between 1 and 99.
BYTERCVD	Bytes received
BYTESENT	Bytes sent
ENABLED	Specifies if this SAP is enabled. Y for enabled, N for disabled.
ERRCOUNT	Number of errors
FILEMISS	Number of files not found
FILERCVD	Number of files received
FILESENT	Number of files sent
LASTAXSD	Last Date/Time Accessed in ISO format
NAME	Friendly name of the SAP.
OBJRCVD	Number of objects received
OBJSEND	Number of objects sent
PRIORITY	Priority for this SAP obtained from the CLIENT.LOCATION class instance.
PROXY	The internet proxy URI through which the client will connect to the SAP. Maintained by client.
RETRYINT	Number of seconds to wait before retrying the SAP. If blank or non-existent, then there will be no retry.
RETRYLIM	Number of failed attempts, between 0 and 99, before skipping this SAP. If blank or non-existent variable, then there will not be a retry.
ROLE	Role of the SAP. Possible values are: **O**: Client **O**perations Profiles **M**: Self **M**aintenance **S**: **S**ervices **D**: **D**ata **A**: **A**ll of the above roles
SPEED	Speed to the SAP from the client computer measured in Bytes per second
STATUS	Status of this SAP 000= SAP was accessed successfully 920 = SAP could not be accessed 999 = SAP was not used

Table 9.5 ~ SAPSTATS Object Attributes

Attribute	Description
STREAM	Specifies if streaming is used. Y for enabled. This overrides the client setting in ZMASTER.ZNORSPNS.
THROTYPE	Type of bandwidth throttling used. Possible values are NONE, ADAPTIVE, and RESERVED.
TIMEOUT	Communications timeout in seconds.
TYPE	Type of SAP. Possible values are: **RCS** - Radia Configuration Server **DATA** - Radia Proxy Servers, Radia Staging Servers or a CD-ROM.
URI	Universal Resource Identifier for the SAP

Radskman Execution (PREFACE)

The PREFACE object contains information about each execution of **radskman**. PREFACE is sent to the Radia Configuration Server at every phase of a **radskman** process including:

■ Client Operations Profile resolution

■ Client Self Maintenance resolution

■ Catalog resolution

■ Single service resolution (This can happen multiple times depending on what services are processed.)

■ Outbox flush

At each new phase of the Client Connect, the PREFACE object is updated. The variables in the PREFACE object can be used for resolution and reporting. For resolution, use the attributes of the PREFACE object for ZSTOP expressions, for symbolic substitution, and for dispatching messages. For reporting, you can combine MACHNAME, ZUSERID, ZCONTEXT, JOBID, and CTYPE to know which user ran the Client Connect, the type of connect, and the context.

Table 9.6 ~ PREFACE Object Attributes

Attribute	Description
CMDLINE	The **radskman** command line parameters used for the current Client Connect.
COMPDN	The distinguished name of the computer in the Active Directory format. This field will be blank if the system is not part of an Active Directory or a Domain environment. Windows operating systems that do not authenticate to Active Directory would show this as their DomainName/MachineName. **Example**: CN=ALEE,CN=Computers,DC=usa,DC=asdfoods,DC=com

Table 9.6 ~ PREFACE Object Attributes	
Attribute	**Description**
CTYPE	Type of Radia Client. The possible values are:
	RSM Radia Software Manager
	RAM Radia Application Manager
	RPS Radia Proxy Server or Radia Staging Server (for preloading application resources)
JOBID	The jobid specified on the command line for this connect (Radia Client versions 3.0 and above).
LOCALUID	The starting directory under IDMROOT on the client computer. LOCALUID contains the value derived from the STARTDIR **radskman** parameter. For example, if startdir = $USER then LOCALUID would contain the user's ID. If STARTDIR = SYSTEM then LOCALUID would contain 'SYSTEM'. UID stands for user's initial directory *not* the user's identification.
MACHNAME	Client computer's machine name.
USEREXEC	The user who is currently logged on and who executed the command. For Notify and Timers, this would be SYSTEM. For logon scripts, this would be the subscriber's network account name.
ZCONTEXT	The context for this connect. Possible values are: M – Machine, U – User, or blank.
ZDOMNAME	The Radia Database's domain specified in the DNAME parameter of the **radskman** command line. If DNAME is not specified in the command line, the default is SOFTWARE.
ZMGRNAME	The Radia Configuration Server's name specified in the MNAME parameter of the **radskman** command line.
ZUSERID	The ZUSERID field contains the same value found in ZMASTER.ZUSERID of the client. In most scenarios, this represents the machine name of the client computer, but may also contain the current user name or another value. The value found in this field is often used as the key for policy resolution or reporting. The UID **radskman** parameter sets this value.

Systems Management Information (SMINFO)

The Systems Management Information (SMINFO) object is created on all Radia 4.x client computers. The SMINFO object summarizes hardware specific information that is independent of what operating system or software is installed on the client computer. Radia uses SMBIOS standards to access data about the BIOS. SMINFO also includes some network and user ID information.

Table 9.7 ~ SMINFO Attributes	
Attribute	**Description**
ASSETTAG	Unique Asset Tag number of the client computer from the BIOS
COMPDOMN	Computer Domain
COMPNAME	Computer Name

Table 9.7 ~ SMINFO Attributes

Attribute	Description
IPADDR	Client computer's IP address
MACADDR	Client computer's MAC address
MACHUUID	Unique machine user ID
SNENCLOS	Serial Numbers for the system enclose structures from the BIOS
SNSYSTEM	Serial Numbers for the system structures from the BIOS
SUBMASK	Subnet Mask
SUBNET	Subnet
SYSMANUF	System manufacturer from the BIOS
SYSPROD	System manufacturer product information from the BIOS

The PROFILE File

Some client objects such as ZCONFIG and ZMASTER are sent to the Radia Configuration Server during a Client Connect. The objects received from the client computer are stored in the PROFILE file in the Radia Database. Within the PROFILE file, each client computer is stored as a domain. By default, each client computer is identified by the subscriber who is currently logged on. The subscriber may be either a computer name or a user name.

The following table describes some of the objects that you might find in the PROFILE file, although the objects may vary based on your configuration.

Table 9.8 ~ Objects in the PROFILE File

Instance	Information Recorded
ZCONFIG	Contains basic hardware information for the client computer such as processor, operating system, and drives.
ZMASTER	Contains information used to run the Radia Clients, such as user ID and operating system.
ZSVCSTAT	Contains information about the service after it has been successfully installed on the client computer. This is useful for reporting purposes such as determining which users have the application or when it was installed. One instance is created for each service.
ZSTATUS	Contains information about the most recent Client Connect, such as the number of objects going to and from the client computer.

Each domain contains several classes, which represent the objects received from the client computer. Use Radia System Explorer to view the PROFILE file.

Reporting with Radia Inventory Manager

The Radia Inventory Manager is a policy-driven, inventory management tool that automatically discovers information about software and hardware, and consolidates the results into Web-based reports. The Radia Inventory Management client is a WBEM (Web-based Enterprise Management) consumer. WBEM is a standard for collecting information defined by the Distributed Management Task Force (DMTF). For additional information on WBEM, see http://www.dmtf.org.

The Radia Inventory Manager consists of two parts, the Radia Inventory Manager server and the Radia Inventory Manager client. The Radia Inventory Manager server enables centralized reporting and administration based upon the discovery results. The Radia Inventory Manager client discovers configuration information on the client device, and reports the results to the Radia Inventory Manager server. The results are sent to the Radia Configuration Server during connection activities, and forwarded to the Radia Integration Server. Optionally, the data can be sent for insertion into an ODBC-compliant database. You can view the Radia Inventory Manager reports from either the Radia Integration Server or from the Radia Management Portal, if installed.
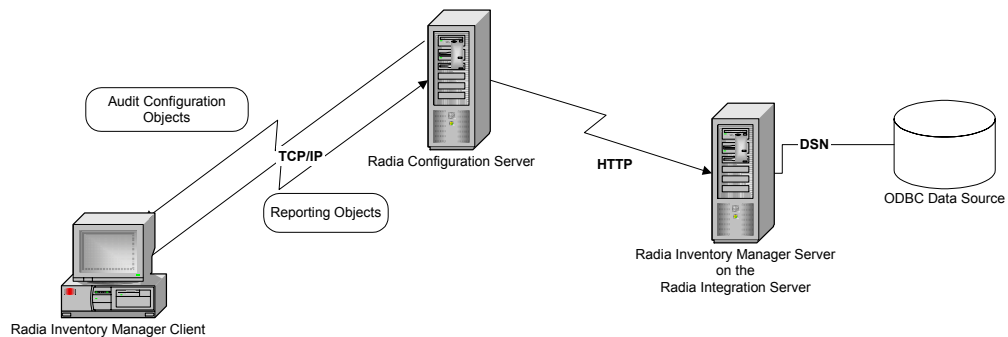
Figure 9.4 ~ The Radia Inventory Manager communicates with an ODBC source.

The Radia Inventory Manager provides the following benefits:

■ Audits hardware information.

■ Audits software, including the ability to audit specific files or classes file, such as .EXE files.

■ Collects files from client devices.

■ Generates reports on collected information through an ODBC-compliant database.

■ Reports on Personal Digital Assistants (PDAs).

The Radia Inventory Manager is one module of the Radia Integration Server. The Radia Integration Server stores and maintains the information discovered by the Radia Inventory Manager in an ODBC-compliant database, and uses active server pages to provide reports from

the Radia Database. For computers with WBEM installed, such as Windows 2000, the Radia Inventory Management client can query the WBEM namespace and return information such as hardware, file system, registry, services, and event logs. WBEM client software is available for Windows 95, Windows 98, Windows 98 SE, and Windows NT4 SP4 or higher from Microsoft. For computers that do not have WBEM installed, the Radia Inventory Manager uses Radia Client objects to generate information on the file system and hardware. See the *Radia Inventory Manager Guide* for additional information. For more information, see the *Radia Inventory Manager Guide.*

Important Note

The Radia Inventory Manager is an additional feature available from HP. Contact your HP sales representative for details.

Client Logs

One log is created from the three main client modules, **radskman**, **radpinit**, and **radconct**. The default name for the log is connect.log, and its default location is <System Drive>:\Program Files\Novadigm\log. When connect.log reaches a size of 1 MB, a backup is created called connect.bak.

For diagnosing problems, you may want to name a new log for debugging deployments or for collecting information from your client computers. Each of the three main client modules can be directed to use a particular log file, by adding the **log** parameter to its command line. For example, you might name your logs based on the date and time. Then, if you notice a problem occurring on a certain date, you can retrieve only the logs that you need to review. Add the log parameter to a **radskman** command line to specify a particular log name as shown below:

```
radskman log=notify10012003.log
```

The value for the log parameter is stored in the LOGNAME attribute, located in the ZMASTER object in the catalog and application directories.

Diagnostic Module (radstate)

Radstate is a diagnostic module designed to give an overview of the current state of the Radia Client. The information in the **radstate** output is based on data retrieved from numerous Radia Client objects.

When **radstate** is run with the Verbose parameter (mode **v**), it provides basic information regarding the Radia Client environment:

■ Global object statistics

■ Current date and time

■ Current operating system

■ Locations of the IDMSYS, IDMLIB, and IDMLOG directories

■ Environment settings

■ Emulator settings

■ Trace levels

■ Timeout settings

■ All Service status including component totals by User and Service including instance totals and byte totals.

■ Radia Timer Information

Use **radstate** at anytime to check Radia Client configurations. For example, run **radstate** at the end of each Client Connect. After **radstate** is run using mode option **o**, the ZRSTATE and ZRSTATES objects are built and can be sent to the Radia Configuration Server as needed.

Radstate should be run:

■ Whenever Radia Client-specific information is required.

■ If it is suspected that some files may not have deployed correctly.

■ If desktop updates have not occurred.

Manual execution of **radstate** produces a summary style report, **radstate.log**, regarding the current state of services and resources installed on the client desktop. **Radstate** is executed from the command line using the appropriate parameters, separated by a comma, for example:

```
radstate mode=vo, IDMROOT=C:\Program Files\Novadigm\Lib
```

See the HP OpenView web site for a technote with additional information.

Summary

■ The ZCONFIG object stores hardware configuration information from the client computer.

■ Use the Radia Client Explorer to view objects stored on the client computer.

■ The PROFILE file in the Radia Database stores objects received from the client computer.

■ To generate reports from the Radia Database, consider using the Radia Inventory Manager.

Naming Conventions

This appendix discusses the use of naming conventions to help you organize the software stored in the Radia Database.

When publishing applications, subscribers may have different requirements such as:

■   Different operating systems.

■   Different amounts of free space on their hard drives.

■   Different processors, memory, and so on.

■   Different data or applications, depending upon their job function or other factors.

Due to these varying requirements, you might need to create several packages for a single application. To keep your digital assets organized in the Radia Database, we recommend that you create a naming convention to be used within your organization.

This section provides some recommendations that you can use as a starting point to create your own standards.

Categorizing Information

In general, consider using unique high-level identifiers with an underscore (_) to categorize information in the Radia Database. The Radia System Explorer groups instances based on the identifier that precedes the underscore.

Note
If you decide to use a high-level identifier *without* an underscore (_), you can use the Radia System Explorer's filtering capabilities to display only the instances with that identifier.
See the *Radia System Explorer Guide* for more information.

For example, if you had a Windows 95/98 version and a Windows NT/2000 version of an application to calculate loan amortizations, you might name the packages **AMORTIZE_95/98** and **AMORTIZE_NT/2000/XP** as shown in Figure A.1 on page 439.
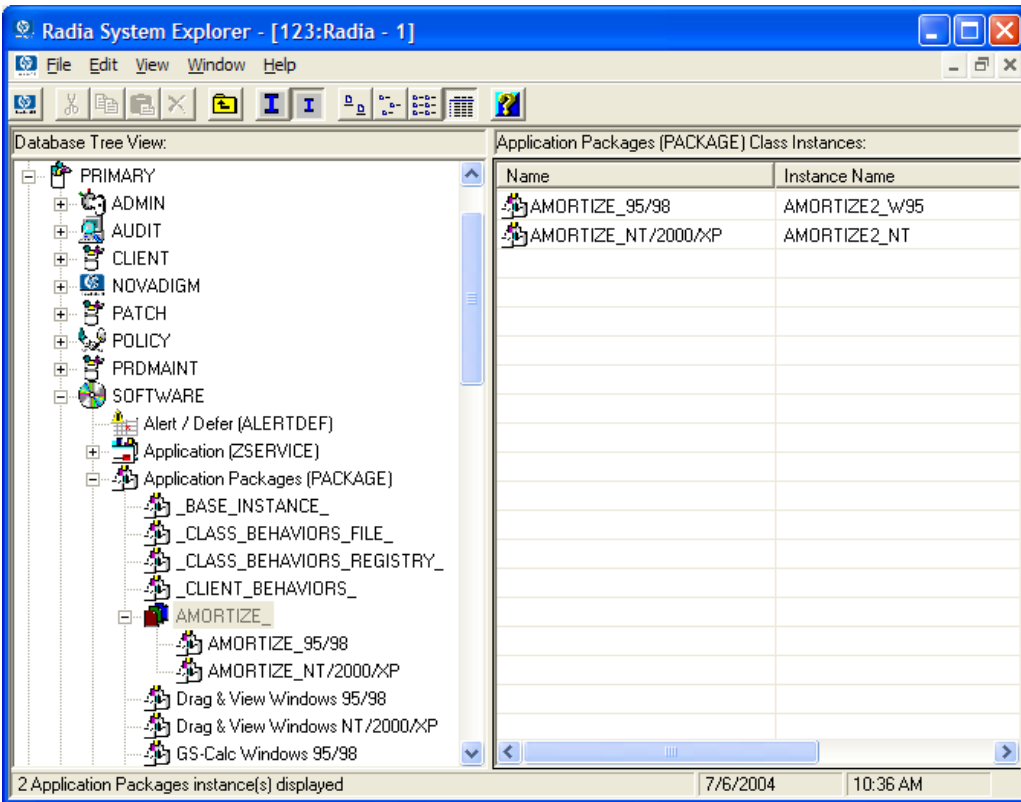
Figure A.1 ~ Instances grouped by identifier.

Naming Conventions for the POLICY Domain

We recommend that you use a variation of the following standards.

Table A.1 ~ Naming Conventions for the USER Class		
Format	**Description**	**Example**
USERID	Identifies the subscriber.	SJones

When naming instances in a workgroup, use information that groups your subscribers appropriately. For example, if your company is organized by division and location, you might use conventions such as the following:

Table A.2 ~ Naming Conventions for the WORKGRP Class		
Format	**Description**	**Example**
DIV_LOC_DESC	Defines ownership or assignment.	CTS_CLE_EVERYONE
• DIV	Identifies the division.	CTS (Corporate Technology Services)
• LOC	Identifies the location.	CLE (Cleveland)
• DESC	Provides additional description of the group.	EVERYONE (all users)

Naming Conventions for the SOFTWARE Domain

In a company organized by division and location, you might organize your digital assets using the following standards.

Table A.3 ~ Naming Conventions for the PACKAGE Class		
Format	**Description**	**Example**
DIV_LOC_APPNAME_VER_OS	Defines the application.	CTS_CLE_WINZIP_80_WNT
• DIV	Identifies the division.	CTS (Corporate Technology Services)
• LOC	Identifies the location.	CLE (Cleveland)
• APPNAME	Identifies the application.	WINZIP
• VER	Identifies the version of the application.	80
• OS	Identifies the operating system that the application runs on.	WNT

Table A.4 ~ Naming Conventions for Delivery and Auditing Classes*		
		All other classes in the SOFTWARE domain.
Format	**Description**	**Example**
REG_DIV_LOC_APPNAME_VER_OS	Defines the application.	NAM_CTS_CLE_WINZIP
• REG	Identifies the region.	NAM (North America)
• DIV	Identifies the division.	CTS (Corporate Technology Services)
• LOC	Identifies the location.	CLE (Cleveland)
• APPNAME	Identifies the application.	WINZIP

Determining the conventions that make sense for your organization may take some time. However, creating a convention up front and communicating it to all of your Radia administrators will keep you organized in the future.

B

Radia Thin Client Support

This appendix describes Radia Client support for thin client computers.

About the Radia Thin Client

The Radia Thin Client runs after a machine reboot when the Radia Notify Daemon automatically launches a Client Connect. Running on reboot guarantees that the disk overlay has been cleared, and that changes made during the Radia connect can be committed. This limits the amount of data written to the disk overlay by external processes. The Radia Client displays a dialog box to the user before running to reduce the amount of activity and data written to the overlay line.

Any valid Notify request will cause the machine to reboot initiating the Client Connect. When the client starts, the RADWORKP object is processed and radskman launches using the parameters specified in RADWORKP. Since thin client computers have limited software and the users are restricted from modifying the local drives, the Radia Notify Daemon is the recommended method to send changes to the client. Additionally, a Timer could be configured to reboot the machine and initiate the process.

The thin client is designed to run in SYSTEM context. If Radskman is launched in a User context, the first local notify encountered will cause the computer to reboot and to restart the process based on the RADWORKP object.

Below is a list of features included in the Radia 4.0 Client for use with thin clients.

- **MACHFREQ** parameter added to radskman. Use this variable to prevent Radia from running every time a client computer reboots. Set this to a positive integer, n, to run a machine connect only if it has been n hours since the last time a machine connect ran. This value ensures that the Radia Client will not run more the once within the specified timeframe to reduce the number of ROM commits. If you set MACHFREQ to 0, the machine connect will run on *every* reboot of a thin client.

- **PRODTYPE** variable added to the ZCONFIG object. PRODTYPE will contain the value "EmbeddedNT" for thin clients. This value is taken from the registry and is used by the OS to identify Embedded OS configurations. You can use this attribute in ZSTOP expressions.

- **DISKFREE** variable honored in the RADSETUP object. This value is used to maintain the amount of free disk space on the thin client's drive. You can state the amount either as percentage of disk space 1-100% or a byte count 101 and up. Consult your operating system's documentation for disk space requirements. The default in the install.ini file is 5 MB. Radskman will verify the amount of free disk space (and where possible the cache overlay usage) before committing any changes. If these values are determined to be below acceptable levels an error will be logged and the commit/reboot sequence will be skipped to avoid ROM corruptions.

- **Radia Notify Daemon** will reboot the machine for any valid notify request (both local and remote). In addition, the Radia Notify Daemon will process the RADWORDP object on startup. This object needs to be created during the install and contains the Radskman command line parameters (CMDLINE) for the Client Connect.

- **Continue** dialog box added after MACHFREQ is checked to verify if the user wants to continue with the Radia connect process. (Default timeout before continuing: 5 minutes.)

- **Reboot timeout** by default is set to 5 minutes. RTIMEOUT is the parameter that controls this in radskman.

■ An **overlay commit** dialog box displays while the operating system is committing to changes to ROM drive. Some thin client machines commit immediately, which can take several minutes, and others perform a quick setup and perform the operation during a computer restart.

Installation

The functionality for the thin client support is built into the Radia Client. You may want to change some of the default values in the install.ini file to accommodate thin client settings. By default, only the Radia Application Manager client is installed.

■ Modify the radskman line to run on the thin client. The default is RADWORKP_CMDLINE = MachFreq=1

■ Modify the RADSETUP.DISKFREE variable. DISKFREE can be either a percentage of disk space 1-100% or a byte count of 101 or above. To do this, add RADSETUP_DISKFREE = 5120000 or RADSETUP_DISKFREE = 2% as shown in *Figure B.1 ~ Modify the install.ini file* below.

```
[Objects]
; Set Radia object attribute values
; A value of _NONE_ will set the attribute to blank
;
ZMASTER_ZDSTSOCK=3464
ZMASTER_ZIPADDR=RCS
;ZMASTER_ZNTFPORT=3465
;ZMASTER_ZNTFYSEC=Y
;ZMASTER_ZTIMEO=240
;ZMASTER_ZTRACE=N
;ZMASTER_ZTRACEL=040
;ZMASTER_ZUSERID=
;ZMASTER_ZVRFYUID=N
;
;PROXYINF_USEPROXY=
;PROXYINF_DISCOVER=
;PROXYINF_PROXADDR=
;
RADWORKP_CMDLINE=MachFreq=1
;
```

Figure B.1 ~ Modify the install.ini file.

After the client is installed, the changes are committed, and the user is prompted for a reboot.

C

Adding Classes and Attributes to the Radia Database

The following procedures show you how to add a class to a domain or an attribute to a class in the Radia Database.

Caution
Be sure to create a backup of your Radia Database.

Adding a Class to a Domain

At a minimum, you will need the following information before you make the changes to your Radia Database:

■ The name of the domain to which you are adding the class.

■ The name of the new class.

■ The name, size, and description of any attributes you want to add to the new class.

To add a new class

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

   The **Radia System Explorer Security Information** dialog box opens.

Note
The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

   The **Radia System Explorer** window opens.

3. Double-click PRIMARY.

4. Right-click on the domain name where you want to add the class. In this example, we will add a new class to the POLICY domain.

5. Select **New Class** from the shortcut menu.

   The following dialog box opens.
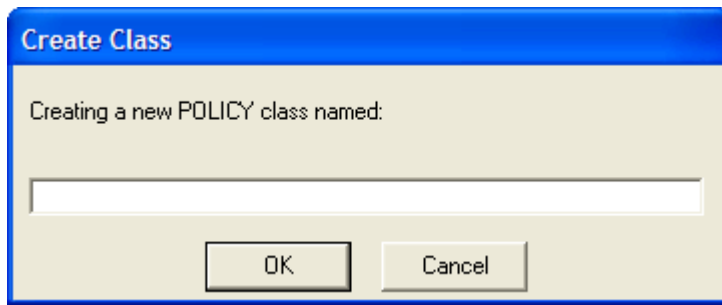
Figure C.1 ~ Create Class dialog box.

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

If the option to open the Class Editor when a new class is added is in effect, the **Editing Class** dialog box will open. The Class Editor allows you to add attributes to the new class as you are creating it. See the *Radia System Explorer Guide* for more information.
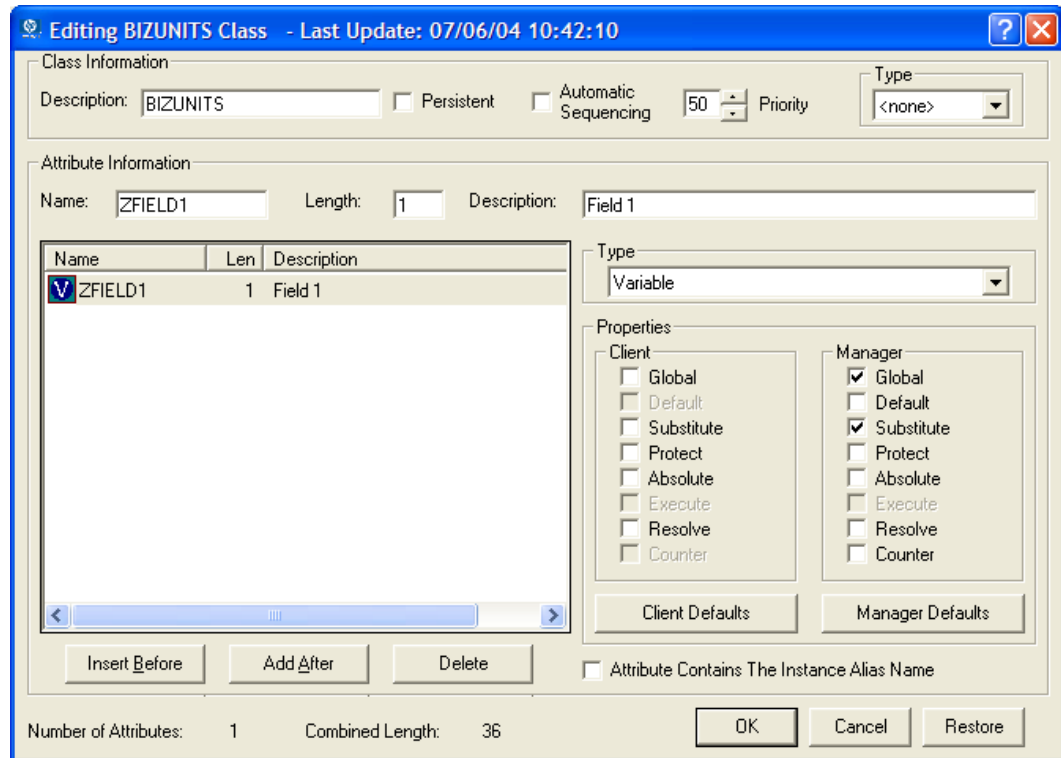
Figure C.2 ~ Editing Class dialog box.

If the option to open the Class Editor when a new class is added is *not* enabled, a new class is added with the specified name, and the Class Editor is not opened.

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

The following message opens.
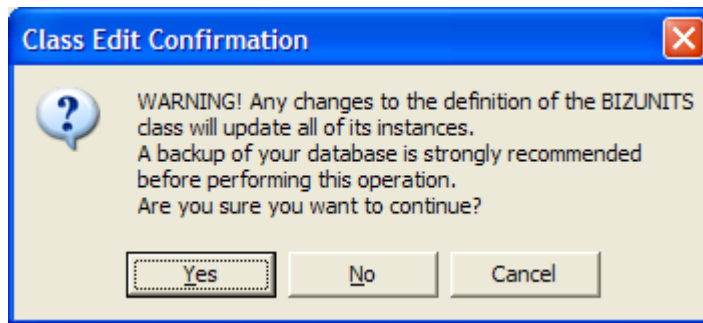
Figure C.3 ~ Class Edit Confirmation dialog box.

8. Click **Yes**,

The new class is added.

Adding an Attribute to a Class

At a minimum, you will need the following information before you make the changes to your Radia Database:

■ The name of the class that you are editing.

■ The name of the new attribute.

■ The length of the new attribute.

■ A description for the new attribute.

To add an attribute to a class template

1. From the **Start** menu, select **Programs**, **Radia Administrator Workstation**, **Radia System Explorer**.

The **Radia System Explorer Security Information** dialog box opens.

> **Note**
>
> The **User ID**, as shipped from HP, is **RAD_MAST**. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** text boxes.

2. If necessary, type a **User ID** and **Password**, and then click **OK**.

The **Radia System Explorer** window opens.

3. Navigate to the class that you want to edit. For example, you might go to **PRIMARY.SOFTWARE.ZSERVICE**.

Note

If you do not see the name of the class (such as ZSERVICE) in the tree view, you can modify the Radia System Explorer options. To do this:

On the **Radia System Explorer** tool bar, click **View**, and select **Options**. In the **Options** dialog box, click the **General** tab, and then select the **Show Class Names Next to Descriptions** check box.

---

4. Right-click the class that you want to edit, such as **Application (ZSERVICE)**.
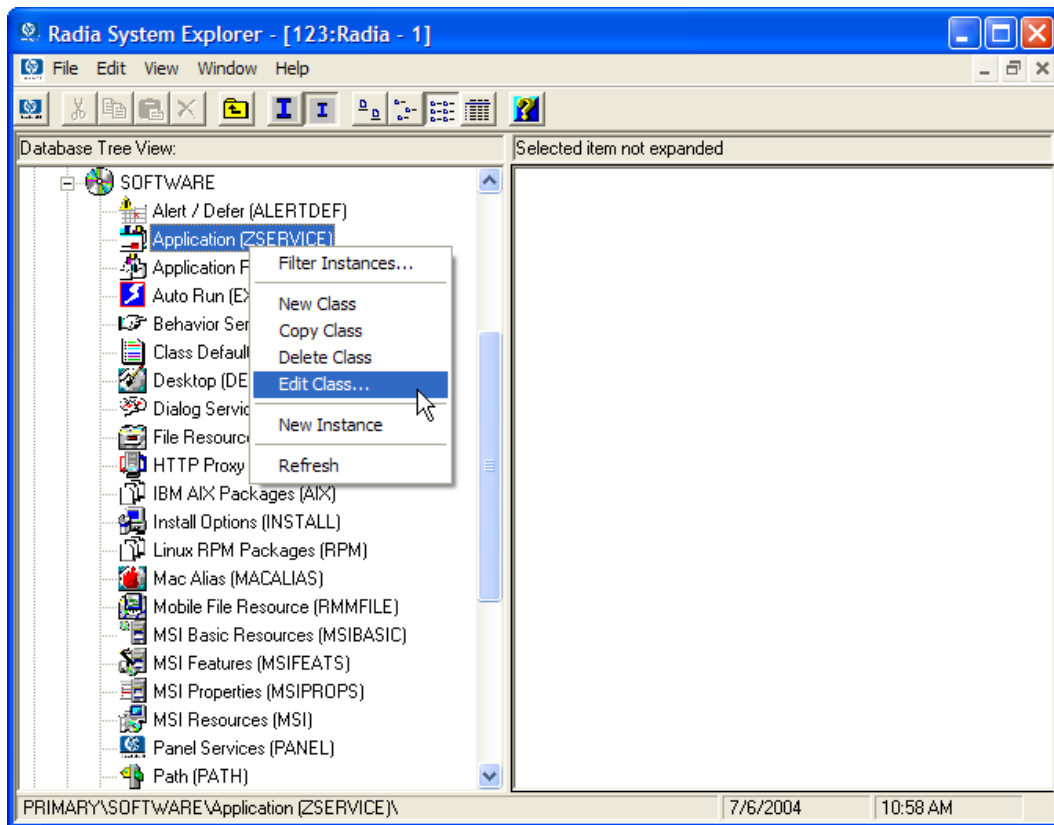
Figure C.4 ~ Edit the ZSERVICE class.

5. Select **Edit Class**.

   The **Editing Class** dialog box opens. For detailed information about the **Editing Class** dialog box, see the *Radia System Explorer Guide*.

6. Determine where, in the **Attribute List**, the attribute should be inserted.

7. In the list of attributes, select the attribute adjacent to where you want to insert the new attribute.

> ### Note
>
> If the **Automatic Sequencing** check box is **not** selected, attributes are processed during resolution in the order in which they appear in the Attribute List.
>
> If the **Automatic Sequencing** check box is selected, the attributes of the class are processed in the following order: Expressions, Attributes, Classes (Connections), and then Methods.
>
> See the *Radia System Explorer Guide* for more information.
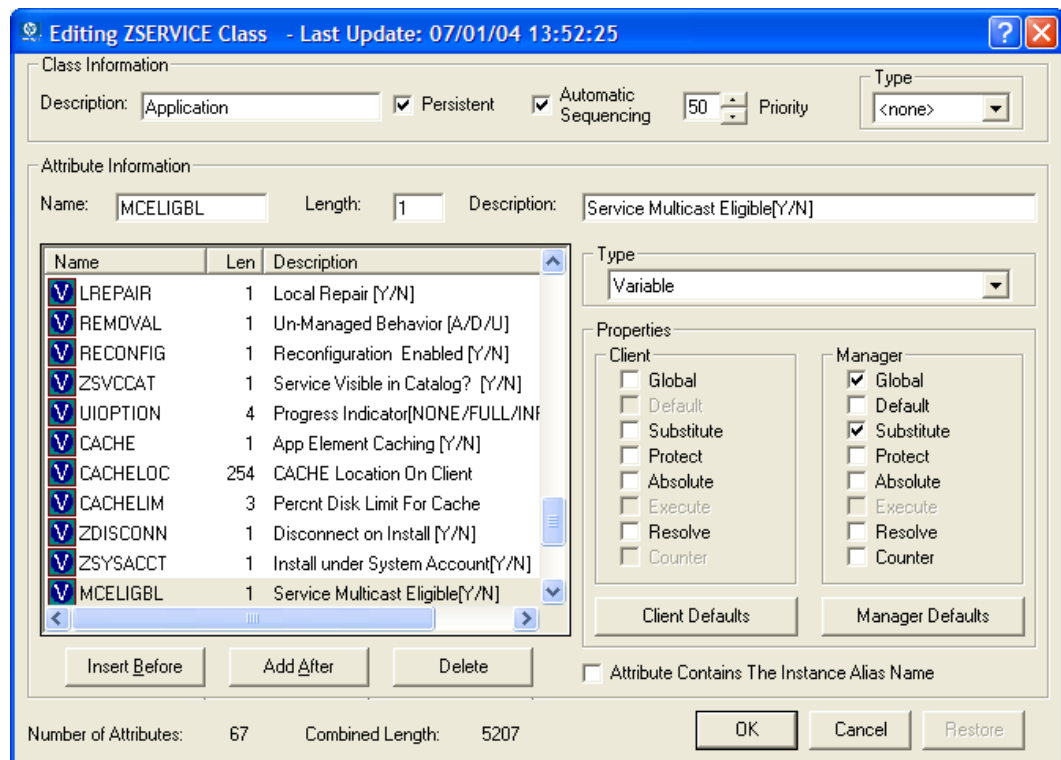
Figure C.5 ~ Select an attribute in the Editing Class dialog box.

---

8. Click **Insert Before** to insert the attribute before the selected one.

   OR

   Click **Add After** to add the new attribute after the selected one.

   A blank attribute appears.
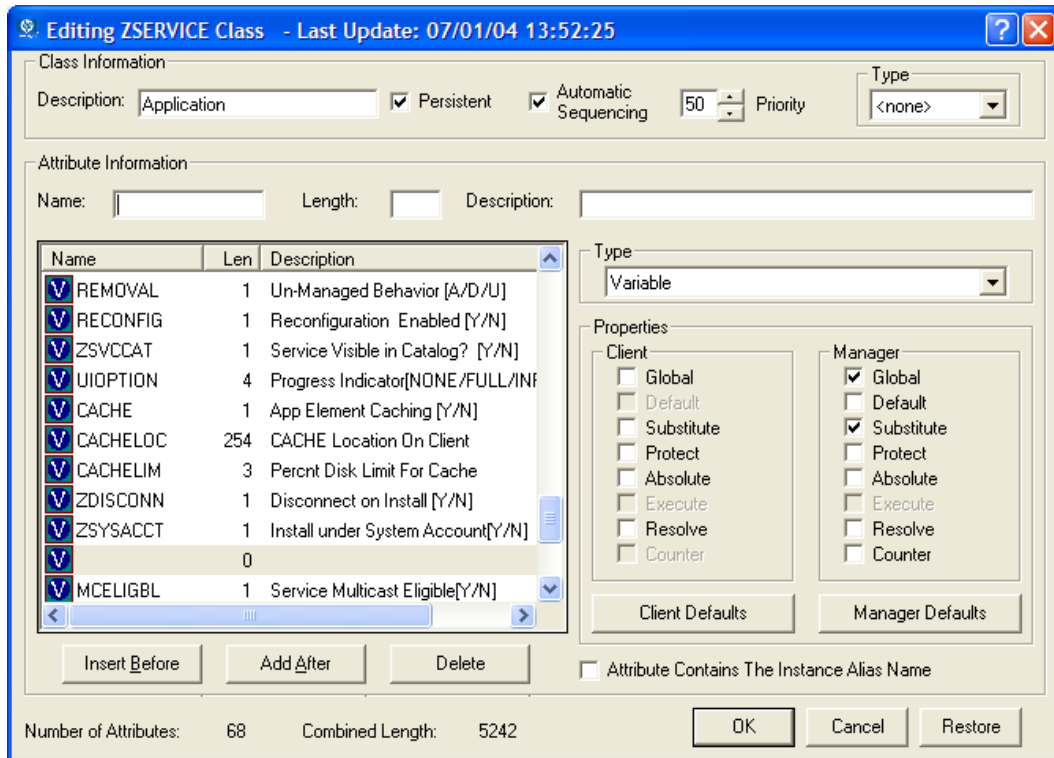
Figure C.6 ~ A blank attribute.

9. In the **Name** field, type the name of the new attribute.

10. In the **Length** field, type the length for the attribute.

11. In the **Description** field, type a description for the attribute.

12. In the **Type** drop-down list, select **Attribute**.
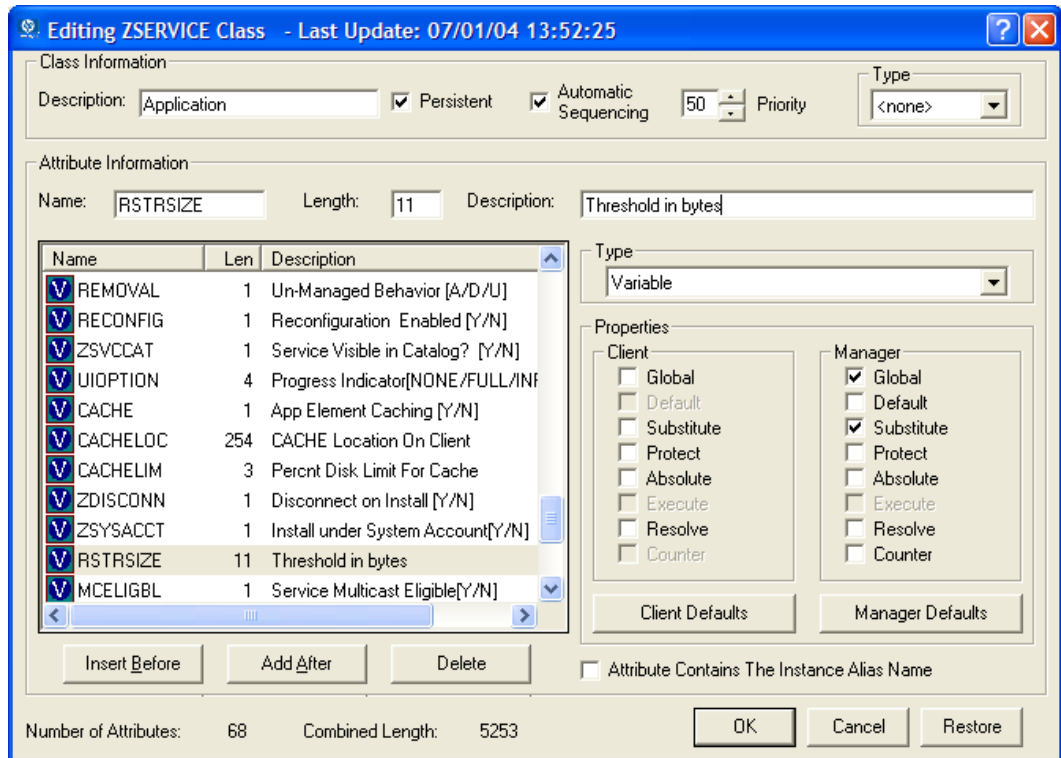
Figure C.7 ~ New attribute information.

13. If the document that contains the information about the new attribute specifies **Client** or **Manager Properties**, select the appropriate check boxes.

14. Click **OK**.

15. Click **Yes** to confirm the changes to the class.

Glossary

Administrative Installation Point (AIP)

Also called *Administrative Control Point (ACP)*.

An **AIP** is a server share or local directory structure that contains all of the files that you may need to run setup for a Windows Installer-enabled application.

Application Manager

See Radia Application Manager.

applications

Also called *software*, *data,* or *services*.

Applications are one type of content that Radia can manage on subscriber computers. Use the Radia Publisher to create packages of data to be managed on your subscribers' computers.

attended

An **attended** computer is a computer that a user is logged on to and using.

attribute

Also called *field, attribute,* or *property*.

An **attribute** is a single, descriptive data item in a class. The class template contains a definition (e.g., the name, data type, description, and length) for each attribute that makes up the class. Class instances contain a set of attributes and each attribute contains a value. attribute property

An **attribute property** controls some aspect of how an attribute is processed on the Radia Configuration Server and client computer. Each attribute defined in a class template has a set of Radia Configuration Server properties and a set of client properties.

audience list

An **audience list** is a directory of the subscribers for an application used by Radia Notify.

base instance

The **base instance** contains the default values for the attributes that make up a class. When you create a new instance in that class, the attributes in the new instance inherit the default values, as specified in the base instance.

byte level differencing

Byte level differencing is the process of publishing a patch containing updates or corrections to a resource. The patch is calculated by differencing an existing copy of the resources in the Radia Database against the resources currently being published.

class

See *instance*.

class connection variable

A **class connection variable** determines the path of resolution for a client's distribution model during the Client Connect process. It is a branch in the resolution process.

A class connection is resolved and resolution continues using the target instance identified in the class connection variable if the class connection variable attribute's name is _ALWAYS_, INCLUDES, REQUIRES, or if the name of the attribute matches the current value of the system message.

class instance

See *instance*.

clean computer

A **clean computer** is a computer on which the operating system has just been installed, and no further changes have been made.

client

The **Radia Client** (Radia Application Manager or Radia Software Manager) runs on the client computer. It communicates with the Radia Configuration Server to receive information about the desired state of the client computer, and compares that information to the actual state of the client computer. Then, the Radia Client makes any adjustments necessary to make the actual state match the desired state.

client computer

A **client computer** is a client computer that has the Radia Client software installed on it.

Client Explorer

See *Radia Client Explorer*.

client object

A **client object** is a file located on the client computer that contains information about the configuration of services or hardware.

component class

A **component class** is a type of class used to identify the items (files, registry entries, links, icons, and so forth) that make up the content identified by a Configuration class instance. Typically, this class' instances have distributable data associated with them such as FILE, REGISTRY, or DESKTOP.

Use the Radia System Explorer's Class Editor to set the class type to "Component".

configuration class

A **configuration class** identifies content to be managed on subscribers' computers by grouping together instances of component classes. Typically, a configuration class' instances do not have distributable data associated with them. They are connected to instances of one or more component classes, perhaps through an instance of another configuration class. Examples: ZSERVICE, PACKAGE, VGROUP, VERSION, and so forth.

Use the Radia System Explorer's Class Editor to set the class type to "Configuration".

desired state

The **desired state** embodies the content that Radia manages for a specific client computer. A model representing the desired state for each client computer is stored in the Radia Database. The desired state model is created and managed using the Radia System Explorer.

domain

A **domain** logically partitions a file in the Radia Database to group "like" classes together.

Examples: POLICY domain, SOFTWARE domain, SYSTEM domain.

- The POLICY domain contains the classes that identify users individually and by their association with groups of other users.
- The SOFTWARE domain contains the classes needed to define and deploy applications. Radia administrators will do most of their work in the POLICY and SOFTWARE domains of the PRIMARY file.
- The SYSTEM domain contains the classes that contain administrative and process control definitions.

expression variable

An **expression variable** contains a single line REXX command that is executed during resolution. If the expression evaluates to "true" in an attribute named ZSTOP, it causes resolution of the current instance to end. Resolution continues in the calling instance with the variable following the one that called the instance containing the expression variable.

file

A **file** is the highest level in the hierarchy of the Radia Database and it groups similar domains together.

Example: PRIMARY file.

The PRIMARY file is used to define and maintain the distribution model. This is one of the pre-configured files distributed with the Radia Configuration Server and installed when you first install Radia. Others are the NOTIFY file and the PROFILE file. Radia administrators will do most of their work in the PRIMARY file.

instance

Also called *class instance*.

An **instance** is a Radia Database object containing a specific occurrence of a class. This is analogous to a row in a relational data table or a record in a traditional flat file. The attributes of an instance contain the data describing one specific entity of that class.

Inventory Manager

See *Radia Inventory Manager*.

Manager

See *Radia Configuration Server*.

mandatory service

A **mandatory service** is a service that is required on the client computer. Services are made mandatory by setting the ZSVCMO variable in the Application instance to **M**.

method

A **method** is a program that performs functions that are meaningful in the context from which they are called.

Methods can be written in REXX or in a language that produces an executable that can validly run on the platform where it is invoked. The REXX run-time environment that we provide interprets REXX methods.

Client methods run on the client computer, while Radia Configuration Server methods run on the Radia Configuration Server computer.method variable

The **method variable** identifies the method, or program, to be executed as part of the resolution process.

For Radia Configuration Server methods, it contains a reference to an instance of the SYSTEM domain PROCESS class that identifies the method to execute and the parameters to be passed to the method. Radia Configuration Server methods are located in the Radia Configuration Server BIN subfolder for .exe methods or in the Manager REXX subfolder for REXX methods.

For client methods, it contains the name of the method to execute on the client computer. The name of a method variable that executes a client method identifies the event (such as installing or removing software) for which the method should be executed. Client methods are located in the IDMSYS location on the client computer.

Notify

A **notify** forces one or more client computers to connect to the Radia Configuration Server to install, update, or remove an application, or send an e-mail to subscribers of a particular service.

null instance

The **null instance** of a class is used when an instance of that class that does not exist. During resolution, if a connection to a non-existent instance of a class is attempted, the null instance is used. This provides a resolution path that handles broken connections.

object

An **object** is a data structure containing variables stored in a file with an .EDM extension on the client computer. An object can consist of one or more instances. Each instance contains the same set of variables. The values held in the variables can vary from instance to instance.

Use the Radia Client Explorer to view, edit, or create objects.

optional service

An **optional service** is a service that is available to subscribers via the Radia Software Manager user interface. Services are made optional by setting the ZSVCMO variable in the Application instance to **O**.

package

A **package** is the data that is published as an individual unit.

policy

A **policy** determines *which* subscribers (or computers) have access to *what* software. The POLICY domain class instances identify users. Connections to the POLICY class instances identify the content to be managed for those subscribers.

promote

When you **promote** a package that was created with the Radia Publisher, you are storing the package in the Radia Database.

publish

To bundle a set of related data into a single unit that can be managed by Radia.

Radia Application Manager

The **Radia Application Manager** (RADIA_AM.EXE) is the Radia Client that manages mandatory services. The Radia administrator uses the Radia System Explorer to specify the services that the Radia Application Manager manages on the client computer. No user interface is available.

Radia Client Explorer

The **Radia Client Explorer**, installed with the Radia Administrator Workstation, can be used to view or edit local objects, or create new objects. You can also use the Radia Client Explorer to view objects located on a file server or on other computers to which you are connected via a local area network (LAN).

Radia Configuration Server

Also called *Active Component Server* or *Manager*.

The **Radia Configuration Server** distributes applications to client computers. It runs on the server and maintains the Radia Database, which stores information that the Radia Configuration Server needs to manage digital assets for distribution to client computers.

Radia Database

The **Radia Database** stores all of the information necessary to manage digital assets on a client computer, including:

- The software and data that Radia distributes.
- The "desired state" of each client computer with respect to the Radia-managed content.
- The policies determining which subscribers can subscribe to which packages.
- Security and access rules for Radia administrators.

Use the Radia System Explorer to manipulate the Radia Database.

Radia Inventory Manager

The **Radia Inventory Manager** is a policy-driven, inventory management tool that automatically discovers information about software and hardware, and consolidates the results into Web-based reports. The Radia Inventory Management client is a WBEM (Web-based Enterprise Management) consumer.

Radia Publisher

The **Radia Publisher** is used to create packages of data and store them in (i.e., promote them to) the Radia Database.

Radia Proxy Server

When used in your environment, the **Radia Proxy Server** can reduce the load on your Radia Configuration Servers by distributing requested resources to client computers. The Radia

Proxy Server maintains multiple data caches containing the resources needed for each client computer's desired state.

Radia Scheduler

The **Radia Scheduler** service (radsched.exe), installed with the Radia Application Manager, allows you to deploy a service at a specific time.

Radia Screen Painter

The **Radia Screen Painter**, installed with the Radia Administrator Workstation, is used to develop custom dialog boxes.

Radia Software Manager

The Radia Software Manager (radia_sm.exe) is the Radia Client used to manage optional services. The Radia administrator uses the Radia System Explorer to specify the services that are available to the subscriber.

The subscriber installs and manages data that is available from the Radia Software Manager user interface.

Radia Staging Server

The **Radia Staging Server** is used to store data required for deploying applications on a computer other than the computer with the Radia Configuration Server.

Radia System Explorer

The **Radia System Explorer**, installed with the Radia Administrator Workstation, is used to manipulate the contents of the Radia Database.

resolution

Resolution occurs when the Radia Configuration Server accomplishes a unit of work in response to a service request. The unit of work is defined by the contents of the Radia Database and parameters included in the service request itself.

In other words, what Radia does depends upon what information is stored in the Radia Database and what information accompanies the request for Radia to perform some action.

For example, the Radia Client Connect submits service requests by sending an object to the Radia Configuration Server. The Radia Configuration Server then performs resolution in response to each request. The parameters that control the processing of the service request are in the input object.

resource

Also called a *file*.

A **resource** is a single component that is bundled into a package. Examples of resources are files, desktop links, and sets of registry keys.

Scheduler

See **Radia Scheduler**.

Screen Painter

See **Radia Screen Painter**.

service

Also called a *software application*, *application,* or *software*. A **service** represents a group of related packages that define the content to be managed by Radia.

session

A **session** identifies a packaging exercise in Radia Publisher that results in the creation of one Radia package.

Software Manager

See *Radia Software Manager*.

Staging Server

See *Radia Staging Server*.

subscriber

A **subscriber** is the person who uses Radia-managed applications on a client computer.

symbol

A **symbol** is the name of a variable in global memory, preceded by an ampersand.

symbolic substitution

Radia Database instances and client objects consist of variables that contain values. The value of a variable can contain a specification that refers to the value of another variable. During the resolution process, Radia can 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, one of the _ALWAYS_ connection variables in the SYSTEM.PROCESS .ZMASTER instance of the Radia Database contains the value POLICY.USER .&(ZMASTER.ZUSERID). The reference &(ZMASTER.ZUSERID) refers to the ZMASTER object's ZUSERID variable, which contains the user ID typed into the Radia log-on dialog box on the Radia Client, when the subscriber visits the Radia Software Management Web page. If the user typed in JDOE for the user ID, symbolic substitution would render the effective value of the _ALWAYS_ connection variable as POLICY.USER.JDOE.

The substitution is not permanent, i.e., the value in the Radia Database doesn't change. Only the value in the in-storage object derived from the Radia Database instance for the current resolution process contains the substituted value.

The parentheses are required only if the reference is qualified, i.e., contains a period. If the reference is unqualified, the parentheses are optional.

For example, these symbolic substitution specifications are correct:

```
&(ZMASTER.ZUSERID)
&(ZUSERID)
&ZUSERID
```

and this is incorrect:

```
&ZMASTER.ZUSERID
```

System Explorer

See **Radia System Explorer**.

System File Protection (SFP)

System File Protection prevents protected system files from being replaced. When a program attempts to replace or move a file that is protected, the file's digital signature is checked to determine whether the file is valid or not.

Timer

See *Scheduler*.

unattended

An **unattended** computer is a computer that is not currently in use (attended) by a person.

variable

A **variable** is a piece of named storage that contains a changing 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.

version group

A **version group** is a collection of one or more versions of one application that Radia deploys and manages. Use version groups to roll out a new version of an application to the appropriate subscribers, and activate it upon delivery or at a predetermined time.

Web-Based Enterprise Management (WBEM)

Web-Based Enterprise Management (WBEM) is an initiative from the Distributed Management Task Force (DMTF) to develop standard technologies for accessing management information in an enterprise-computing environment.

Windows Management Instrumentation (WMI)

Windows Management Instrumentation (WMI) is the Microsoft implementation, for Windows platforms, of Web-Based Enterprise Management (WBEM). WMI provides support for WBEM's Common Information Model (CIM).

ZCONFIG

The **ZCONFIG** object contains basic hardware information for the client computer such as processor, operating system, and drives.

ZMASTER

The **ZMASTER** object contains information about the client computer that is necessary to run the Radia Client such as the identity of the subscriber and the IP address of the client computer.

ZSTOP

A ZSTOP expression is used to stop the resolution of an instance based on certain criteria. For example, create a ZSTOP expression to deploy a ZSERVICE instance only to client computers with a particular operations system.

ZTIMEQ

The **ZTIMEQ** object is created, based on information in the Scheduler (TIMER) instance, when a timer is deployed to the client.

Lists

Figures

Tables

Procedures

Index

B

D

H

I

O

P

V