HP OpenView Application Manager Using Radia

for the UNIX operating system

Software Version: 4.1

Installation and Configuration Guide

Documentation Release Date: December 2006

Legal Notices

Warranty

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

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

Restricted Rights Legend

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

Hewlett-Packard Company
United States of America

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

Copyright Notices

© Copyright 1998-2005 Hewlett-Packard Development Company, L.P.

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

Trademark Notices

Linux is a registered trademark of Linus Torvalds.

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

OpenLDAP is a registered trademark of the OpenLDAP Foundation.

Acknowledgements

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

TFTP SERVER
Copyright © 1983, 1993
The Regents of the University of California.

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

OpenSSL License
Copyright © 1998-2001 The OpenSSLProject.

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

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

Support

Please visit the HP OpenView support web site at:

http://www.hp.com/managementsoftware/support

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

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

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

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

To find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

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

Revisions

The version number on the title page of this document indicates the software version. The print date on the title page changes each time this document is updated.

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

Chapter 2: Installing the Radia Clients

- All graphical and non-graphical Radia Client installations have been updated for version 4.1.

Chapter 5: Packaging Applications and Content

- The Radia Publisher for Radia Version 3.x has been renamed the Radia Packager for Radia Version 4.x.

Chapter 7:
Configuring Client Operations Profiles

4.1 Client Operations Profiles is a new feature of version 4.1 for the Application Manager client for UNIX. Client Operations Profiles allow you to create redundancy and fail over capabilities for Configuration Servers and Proxy Servers, control hardware scans, diagnostic settings, and user interface options.

Chapter 9:
Deploying Services

4.1 Page 261, Table 31: 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.

4.1 Page 264, Table 32: The following parameters have been added to radskman for use with Client Operations Profiles: cop, datauri, product, and rcsuri.

4.1 Page 265, Table 33: added the upd parameter, which prevents updates to applications during the Client Connect session.

4.1 Page 273, Table 34: NETAVAIL, RETRYRC, RETRYFLG, RETRYINT, RETRYLMT attributes have been added to the TIMER class to allow for retry if a timer event fails.

4.1 Page 279, Table 35: MONTHLY, MONTHDAY, and STARTUP values have been added to the ZSCHDEF attribute in the Scheduler (TIMER) class.

4.1 Page 307, Radia Self Maintenance: is now supplied to the customer in the form of export decks. The Radia 4.0 clients use the PRDMAINT domain.

Chapter 10:
Radia Client Objects and Directories

4.1 Page 316, Table 41: added a row for SMINFO.

4.1 Page 322, 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.

- Page 311, Table 44: Removed the RETRYINT and RETRYLIM attributes from the SAPSTATS Object Attributes table.

4.1 Page 324, Systems Management Information (SMINFO): new section.

- Page 325, Controlling Default Permissions for Directories and Objects: new section.

Contents

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.
- Be familiar with the requirements for a test environment.

About This Guide

This guide covers the *suggested* implementation for the 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 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 Administrator Workstation.

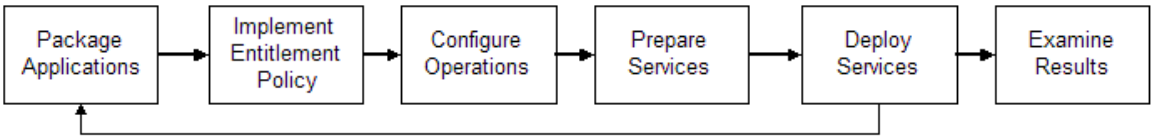


Figure 1: Tasks completed in this guide.

The *Application Manager Guide* covers the following:

- **Installing the Radia Clients**
  This chapter describes how to install the Radia Clients.
- **Installing the Administrator Workstation**
  This chapter describes how to install the Administrator Workstation.
- **Publishing Applications and Content**
  This chapter describes how to publish applications using Component Selection 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 Configuration Servers and 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.
- **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.

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 our solutions automate digital asset management across virtually any kind of network. The following bullets detail each of these distinctive capabilities which 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 what they need, 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, reconfigured – simply by packaging them with other objects or new configuration information. With revisable packaging, value-added service providers and IT administrators can 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 support 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. Radia 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. At a minimum, a Radia distribution model includes the following five elements:

- **Users**
  The identity of the computers being managed.

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

- **Applications**
  The digital assets that are being managed.
- **Application Files**
  The components that make up the digital assets.
- **Deployment Source**
  The location where the application components are centrally stored, such as on a Staging Server or Configuration Server, so they can be deployed to the users.
- **Deployment Destinations**
  The location to which the application and its files will be distributed, such as desktop computers, PDAs, and laptops.

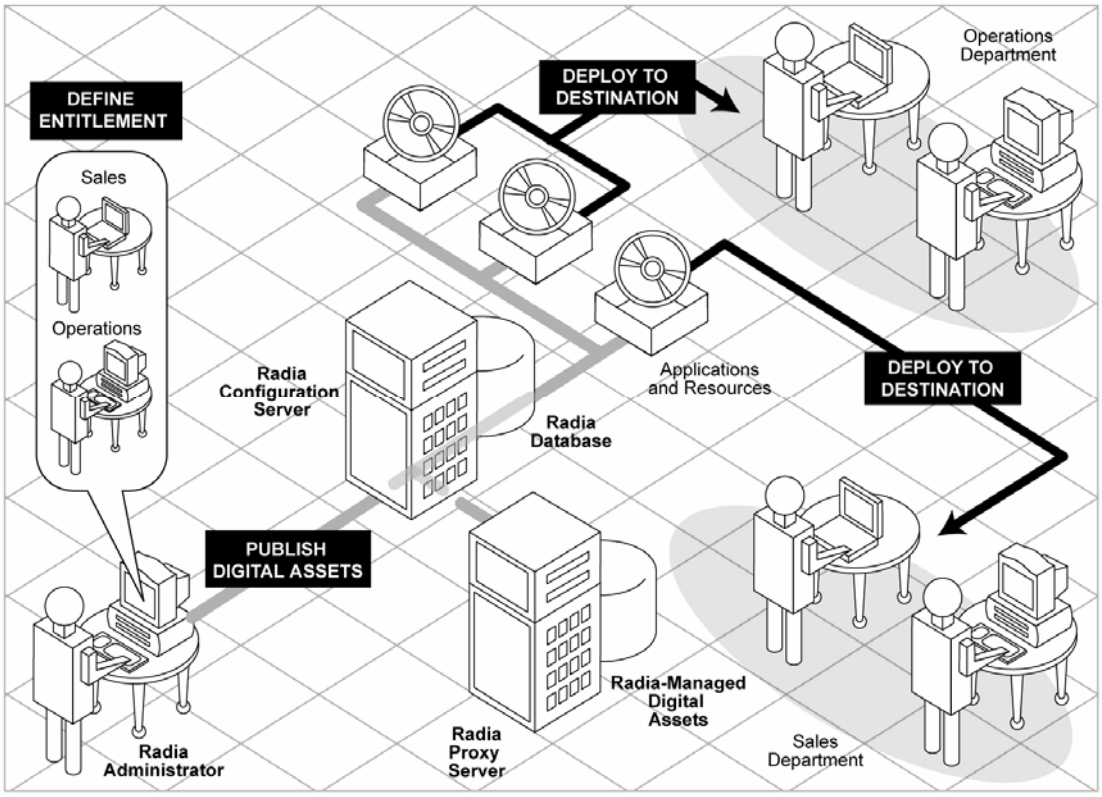


Figure 2: Elements in a distribution model.

Use Radia to manage all of these elements. You will publish **packages** of digital assets, assign these packages to users, and define how the packages will be deployed.

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

The Radia Database

The Radia Database, located on the 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 distribution model for each client computer.
- The policies determining which subscribers are assigned to which packages.
- Security and access rules for Radia administrators.

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

Elements of the Database

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 *Database Reference Manual for the HP OpenView Configuration Server Using 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.

Files and Domains

When you install the 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 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. Refer to the *Inventory Manager Guide* for more information.
  - Use the CLIENT domain to configure Client Operations Profiles. This includes defining which Configuration Servers, Proxy Servers, and Staging Servers the client computer can use. For more information, see Chapter 7, Configuring Client Operations Profiles.
  - 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.

    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. Refer to the *System Explorer Guide*.

  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 Chapter 6, Implementing Entitlement Policy for more information.
- The SOFTWARE domain contains information about the software being managed and the methods used to deploy the software.
- 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 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 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.

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, refer to the *Getting Started Guide for HP OpenView Using Radia (Getting Started Guide)* or the HP OpenView web site.

Configuration Server

The 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 Configuration Server distributes packages based on policies established by the Radia administrator. Refer to the *Configuration Server Guide* for more information.

Management Portal

The Management Portal is a web-based interface that you can use to manage your infrastructure. The 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. Refer to the *Management Portal Guide* for more information.

Proxy Server

If you want to reduce the load on the Configuration Server, or store your digital assets closer to your client computers, consider using a Proxy Server. The Proxy Server stores a copy of the digital assets that are available to subscribers attached to the Proxy Server. The 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 *Proxy Server Guide*.

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:

- **Packager**
  Use the Radia Packager to create groups of components, called **packages**, and promote them to the Configuration Server. See Chapter 5, Packaging Applications and Content for more information.
- **System Explorer**
  Use the System Explorer to view and to manipulate the Radia Database. In addition to this publication, refer to the *System Explorer Guide* for

more information. (Available with the Windows version of the Administrator Workstation).

- **Client Explorer**
  Use the Client Explorer (radobjed) to view and manipulate Radia objects on the client computer. The Radia Client Explorer is installed along with the Radia Packager.
- **Screen Painter**
  Use the Screen Painter to create custom dialog boxes. (Available with the Windows version of the Administrator Workstation).
- **Publisher**
  Use the Publisher to publish Windows Installer files. Refer to the *Publisher Guide* for more information. (Available with the Windows version of the Administrator Workstation).

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 Configuration Server. Install only those clients for which you have obtained a license. The client software is located on the Management Applications CD-ROM.

- **Application Manager**
  Use this client to distribution mandatory applications throughout the enterprise. This client is described in this book.
- **Software Manager**
  Subscribers install, remove, or update optional applications that are available to them in a service list. For more information, refer to the *Software Manager Guide.*
- **Inventory Manager**
  This client allows you to collect hardware information and send it to the Inventory Manager for collection and reporting. Refer to the *Inventory Manager Guide for details*.
- **Patch Manager**
  The Patch Manager analyzes and manages security patches. Refer to the

Patch Manager Guide. (Available with the Windows Radia Client installation).

- **OS Manager**
  The OS Manager controls the provisioning of operating systems. Refer to the *OS Manager Guide*. (Available with the Windows Radia Client installation).

If you install both the Software Manager and 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 Inventory Manager, you can also find out the hardware and software configurations of the client computer.

Summary

- Radia gives you the flexibility and control to manage desktop software efficiently.
- 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 and permissions necessary to deploy the Application Manager.
- Be able to install the Application Manager using either the graphical or non-graphical mode.

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 Configuration Server.

System Requirements

- HP-UX Operating System Version 10.20 or above, PA Risc CPU
- Red Hat Enterprise Linux Version 2.1 and 3.0, Intel Pentium processor or compatible CPU.
- SUSE LINUX versions 8 and 9, Intel Pentium processor or compatible CPU.
- Solaris Operating System Version 2.6 or above, SPARC CPU or Intel Pentium Processor.
- AIX Operating System Version 4.3.1, 5L.
- TCP/IP connection to a computer running Configuration Server.
- Radia client requires 20 MB free disk space.

Prerequisites

- We strongly recommend installing the Radia clients as root. Root authority is required to apply owner and group designators to managed resources.
- Install the Radia client on a local file system.
- The installation program must be run from within UNIX. Although you can continue to work within UNIX (performing other tasks and operations) while the installation program is being executed, we strongly recommend that you don't.
- If you intend to run any of the graphical components of the Radia client software, make sure the UNIX environment variable DISPLAY is set in your environment. If it is not, you will need to set this variable to indicate the hostname or IP address to which you would like to redirect the graphical display.

Table 1: [PROPERTIES] Section of INSTALL.INI

In a…..	Type….
C shell	setenv DISPLAY IP address or hostname:0.0
Bourne, Bash, or Korn shell	DISPLAY=IP address or hostname:0.0 export DISPLAY

If there is an existing installation in the current working directory, you are urged to relocate it before beginning installation. You will be prompted for this during the installation. If you choose to overwrite your existing client, all your customized data will be lost.

When installing the Radia client, you must know the subscribers' operating systems. After setup and configuration, Radia executables and library files will not be changing with the same frequency as that of your site's user files.

To successfully run Radia applications, standard UNIX environment variables are required. Minimally, these environment variables should include the fully qualified path of the installed client executables, the path to the operating system-specific Motif libraries, and the standard UNIX operating system paths for operating system executables and shared libraries. We recommend these be included as part of the logon scripts of the UNIX user ID who installs, and will maintain the Radia clients.

Table 2: Environment Variables

Platforms	Examples
Solaris	`LD_LIBRARY_PATH=/lib:$IDMSYS:$MOTIF:$LD_LIBRARY_PATH` `PATH=/bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`
HP-UX	`SHLIB_PATH=/lib:$IDMSYS:$MOTIF:$SHLIB_PATH` `PATH= /bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`
AIX	`LIBPATH=/lib:$IDMSYS:$MOTIF:$LIBPATH` `PATH=/bin:/usr/bin:$IDMSYS:$MOTIF:$PATH`
Linux	`LD_LIBRARY_PATH=/lib:/usr/lib:$IDMSYS:$LD_LIBRARY_PATH` `PATH=/bin:/usr/bin:$IDMSYS:$PATH`

In Table 2 on page 31, `$IDMSYS` represents the fully-qualified path to the Radia client executables, often referred to as the IDMSYS location. MOTIF represents the fully-qualified path to the Motif libraries installed with the operating system.

The inclusion of the MOTIF libraries is required only when running Radia client or Administrator Workstation graphical tools such as the Packager, the Client Explorer, and the presentation of the Radia client logon panel.

After the Radia client is installed, the file `.nvdrc` is placed in the HOME directory of the UNIX user ID who performed the installation. This file aids you in setting the required environment variables needed to use the Radia clients. We recommend adding a line to the appropriate logon scripts to invoke this shell script:

```
. $HOME/.nvdrc
```

Recommendations

- After you perform an installation, make sure the Application Manager is successfully connected to the Configuration Server. This registers the subscriber in the Radia Database. Once registered, the subscriber appears in the PROFILE file. Make sure to verify that all ports are active and that you have full connectivity to the Configuration Server.

Before you install the Radia client, consider the following:

- You can perform a local installation of the Radia clients.
- Your Radia systems administrator can perform a Remote Installation Setup. This process stores the installation media in a selected directory path. Later client installations can be initiated from any number of intended client workstations providing they have access to the directory path selected during the Remote Installation Setup.
- Performing an installation from a customized configuration file provides a number of benefits.
  - Replication of precise installation details on multiple clients.
  - Ability to use a pre-installation method that runs any script or executable before the Radia client installation.
  - Ability to use a post-installation method, which runs any script or executable after the Radia client is installed.

— You can configure the installation to force a client connection to the Configuration Server immediately after the installation.

— You can pre-configure the IP address and port number of the Configuration Server that the Radia client will be connecting to.

— Ability to use an object update text file that can be used to update Radia objects after the installation.

Installation Methods

You can install the Radia clients by:

- Executing the installation procedure directly from the CD-ROM.
- Copying the files from the CD-ROM into a temporary directory and executing the installation procedure.

Several parameters can be used on the command line when installing the Radia clients. These parameters are used to install the Radia client using the graphical mode, non-graphical mode, plain mode, or silent mode. Table 3 below, describes the installation parameters.

Table 3: Command Line Installation Parameters

Parameter	Example	Description
-mode plain	./install –mode plain	Installs the Radia client in plain mode. The installation graphics are displayed with no animations. This is useful for remote installations where network bandwidth may be an issue.
-mode text	./install –mode text	Installs the Radia client in text mode using the non-graphical installation. The installation takes place completely on the command line. The installation will default to text mode if the DISPLAY environment variable is not set.

Including Maintenance Files with the Client Installation

If additional maintenance files are available, for example, service packs or hot fixes, you can include these files with your client installation by creating a maintenance tar file.

Within your client installation media `/ram` directory, create a file called `maint41.tar` that includes all updated files.

The client installation will check for `maint41.tar` and if found, the client installation will extract all updated files into the IDMSYS directory.

Installing the Radia Client

This section describes both the graphical (using a GUI) and non-graphical (using a command line) installations of the Radia client for UNIX.

In order for Radia to install correctly on HP-UX platforms, you must mount the Radia CD-ROM using pfs_mount.

The Radia CD-ROM is created using the Rock Ridge format. Since the HP-UX standard mount procedure is incompatible with the Rock Ridge file system type, HP has made available the PFS package (Portable File System) that allows their workstations to recognize this format. Specific instructions follow:

Insert the CD-ROM and mount by typing:

```
/usr/sbin/pfs_mount -v -x unix /cdrom/mnt
```

where */cdrom* is your physical CD-ROM device.

To un-mount, type:

```
/usr/sbin/pfs_umount /mnt
```

See your local UNIX systems administrator and UNIX man pages for more information.

Graphical Installation

This section describes how to install the Radia clients both to a local and to a remote computer using a graphical user interface (GUI).

Local Installation

This section describes how to install the Radia clients to a local computer using a GUI.

To install the Radia client to a local computer using a GUI

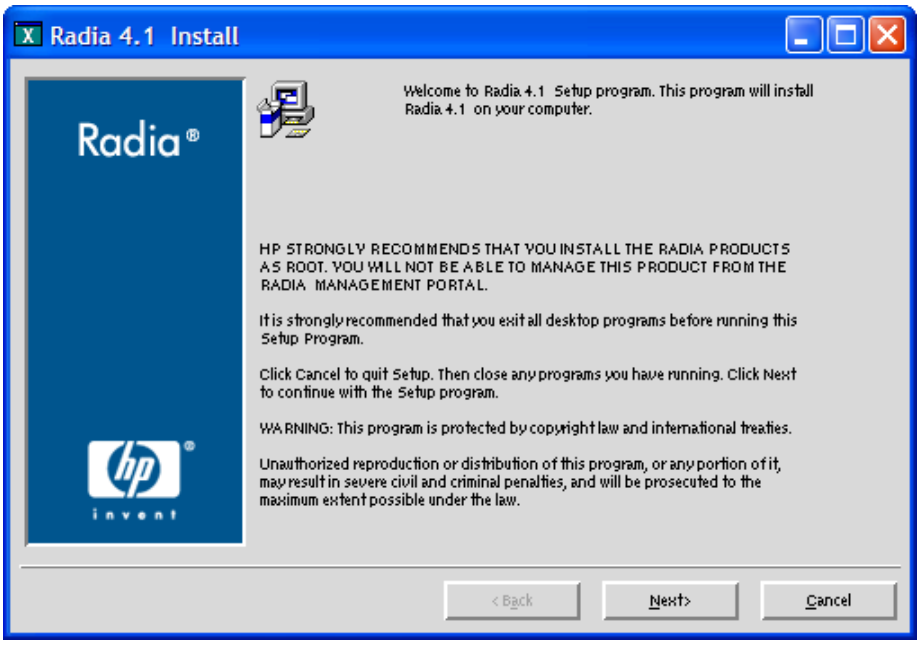
These instructions will guide you through the local graphical installation of the Radia client. For the non-graphical installation instructions, see Non-graphical Installation on page 59.

1 Depending on your version of UNIX, change your current working directory to the correct `/client` subdirectory on the installation media.

Example: For HP-UX, type: **`cd /cdrom/hpux`**

2 Type **`./install`**, and then press **Enter**.

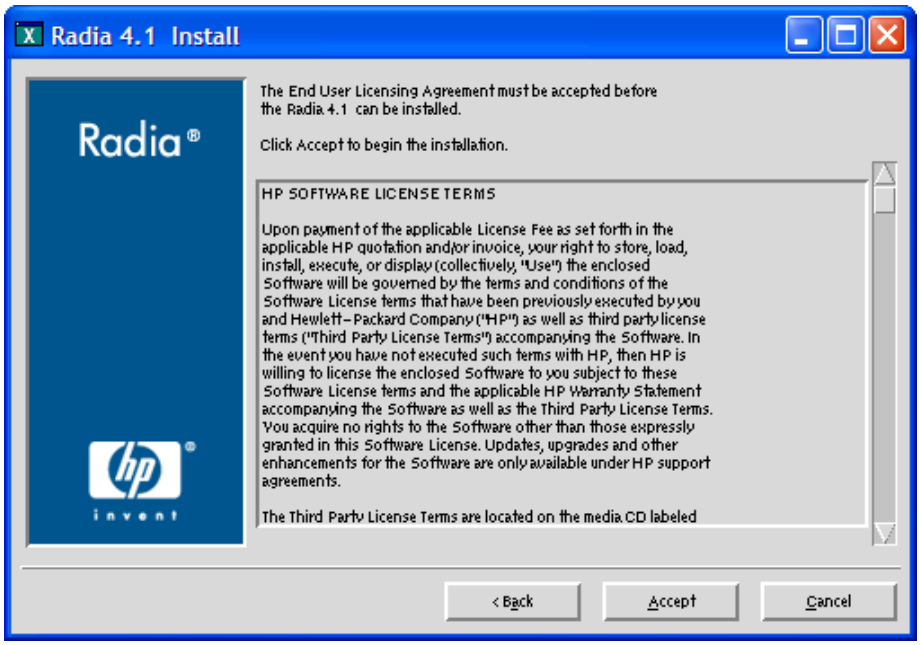
3 The Welcome window opens.



At any point during the installation, you can return to a previous window by clicking **Back**. Also, if you would like to exit the installation at any time, click **Cancel**.

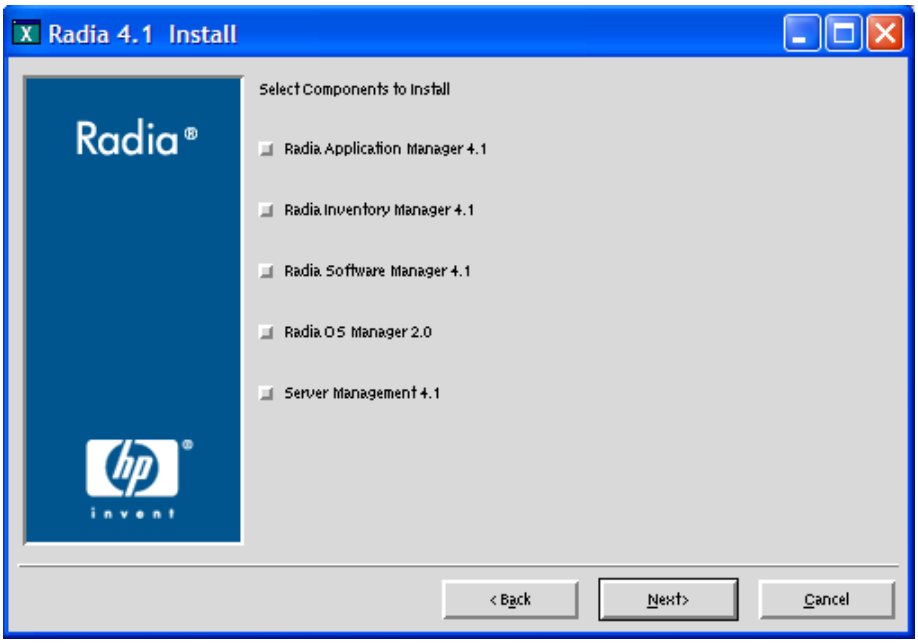
4 Click **Next**.

The End User License Agreement window opens.



5 Read the agreement and click **Accept** to continue.

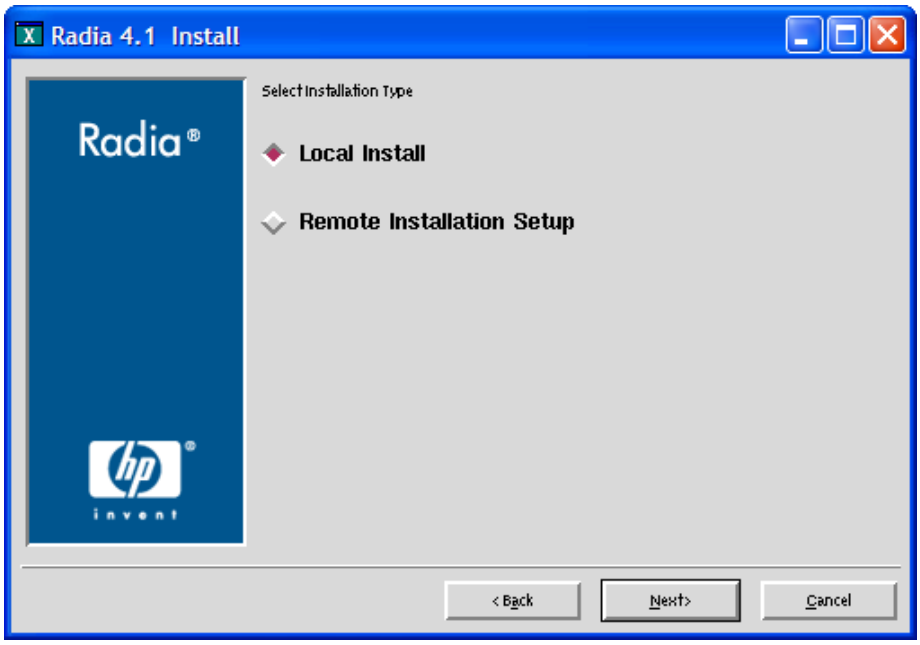
The Select Components to Install window opens.



6 Select the **Application Manager** check box.

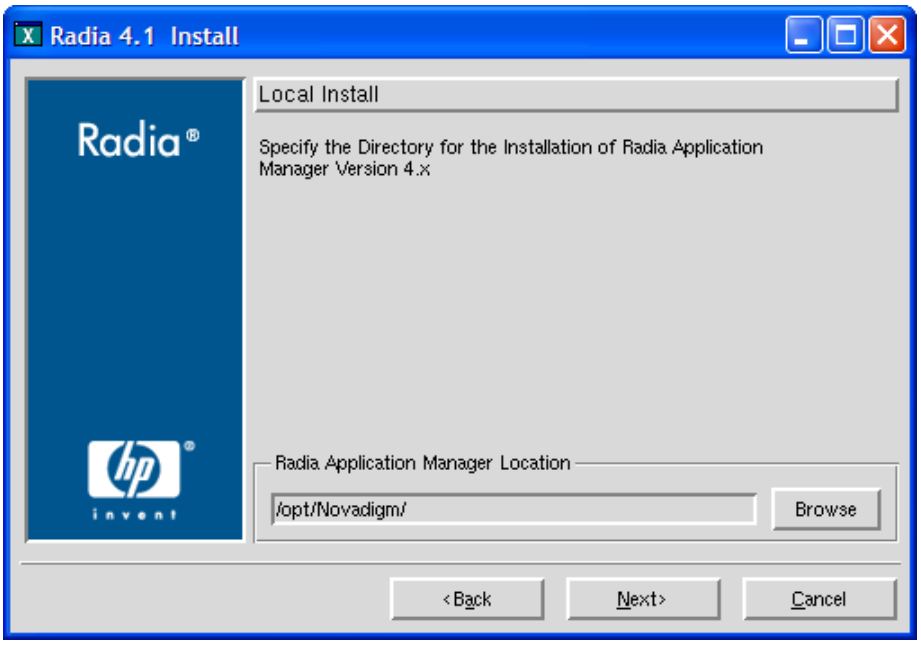
7 Click **Next**.

The Select Installation Type window opens.



8 Select **Local Install** to install the Radia client onto a local computer, and then click **Next**.

The Radia client Location window opens.



9 Type the name of the directory where you want to install the Radia client, or click **Browse** to navigate to it.

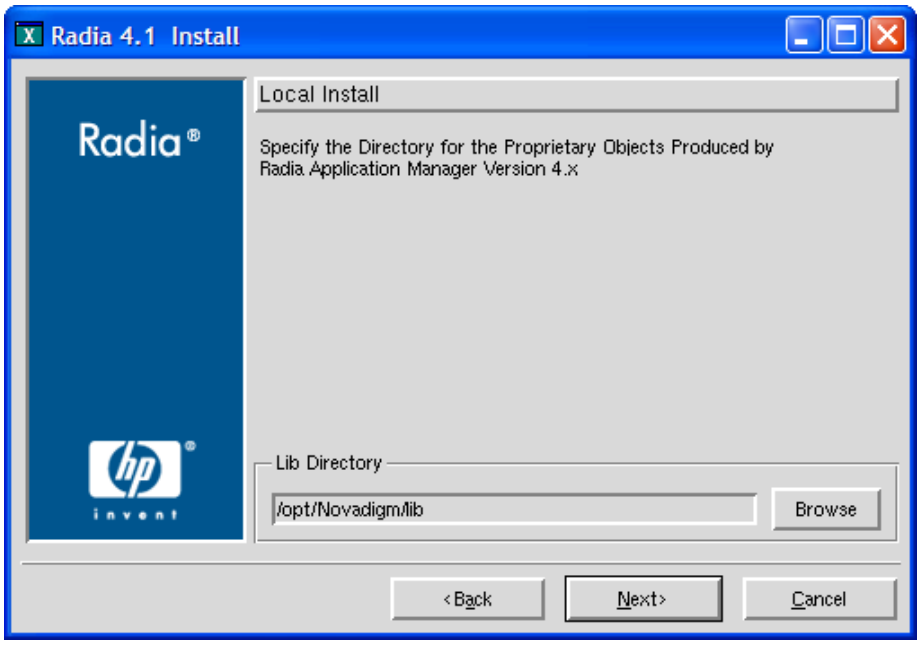
10 Click **Next**.

If the specified directory already exists you will be prompted to verify this location.

— If you would like to update the existing directory, click **OK**.

— If you want to specify a different location, click **Cancel**.

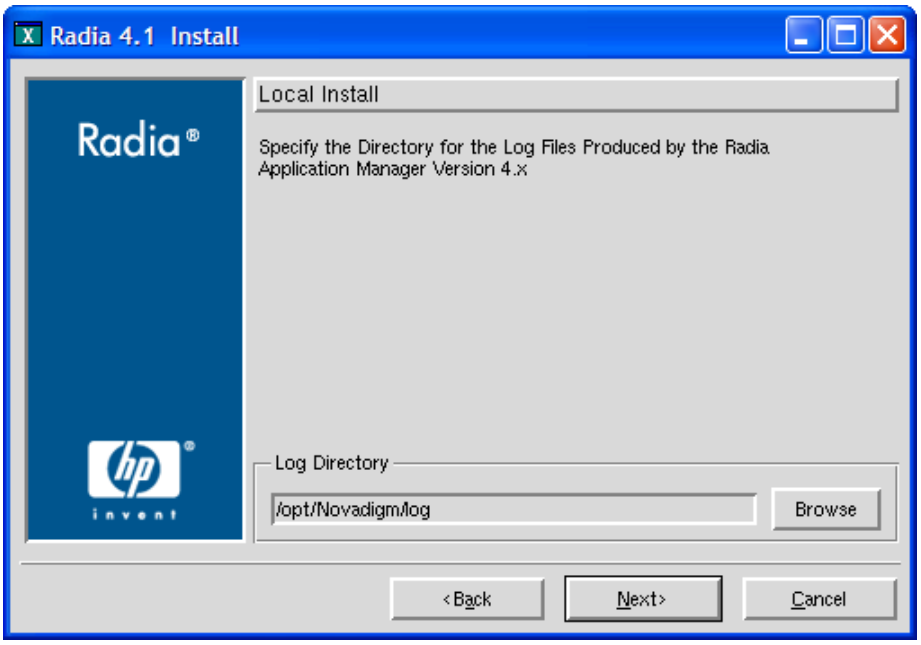
The Lib Directory window opens.



11 Type the name of the directory where you would like to store proprietary information created by Radia (the lib directory), or click **Browse** to navigate to it.

12 Click **Next**.

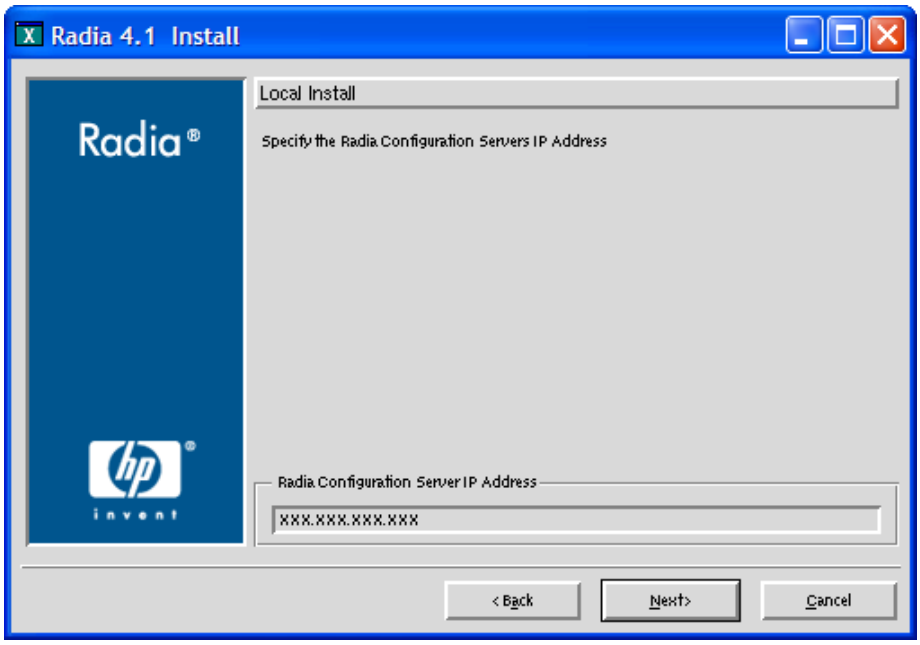
The Log Directory window opens.



13 Type the name of the directory where you would like to store the log files generated by Radia, or click **Browse** to navigate to it.

14 Click **Next**.

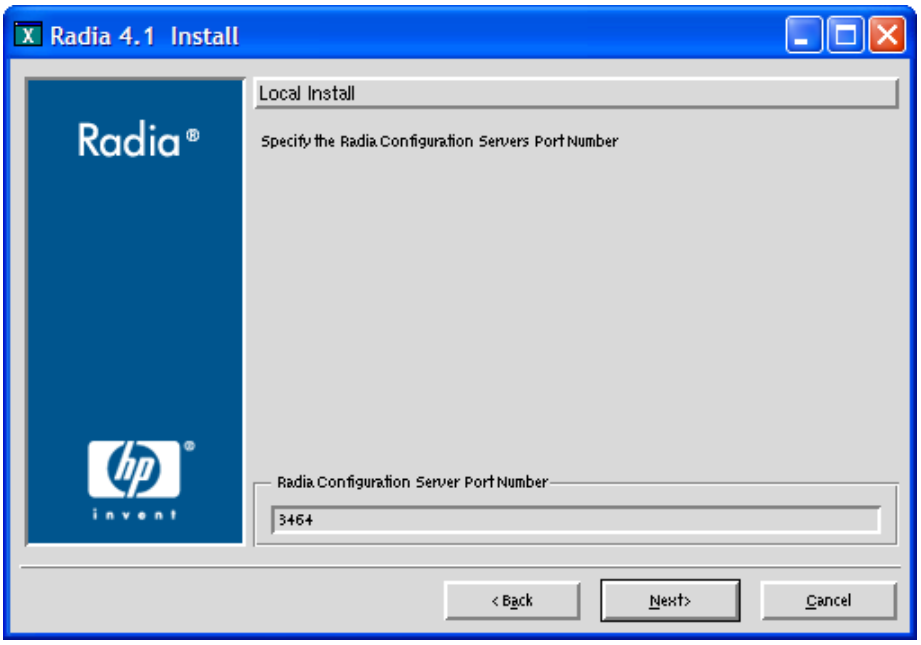
The Configuration Server IP Address window opens.



15 Type the IP address (format: xxx.xxx.xxx.xxx) of the Configuration Server to which the Radia client will connect. Specify a valid IP address or hostname recognized by the client workstation.

16 Click **Next**.

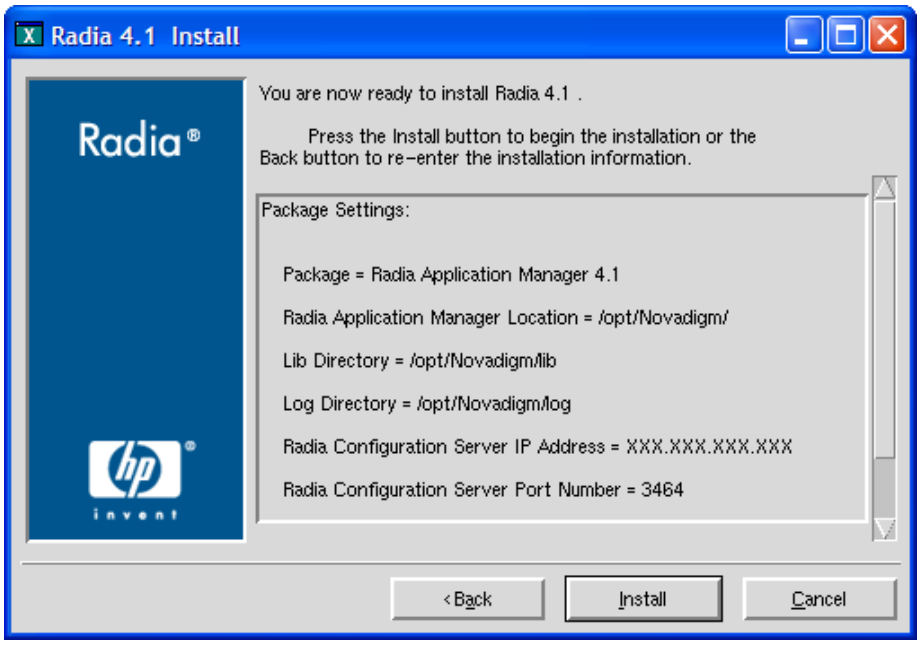
The Configuration Server Port Number window opens.



17 Type the Configuration Server's port number (default is 3464).

18 Click **Next**.

The Package Settings window opens.



19 Review the settings displayed in the Package Settings window. If you would like to change any of the settings, click **Back** until you get to the appropriate window.

20 When you're satisfied with the settings, click **Install** to install the Radia client with these settings.

21 When the installation is complete, click **Finish** to exit the program.

The Radia client has been successfully installed.

Remote Installation Setup

This section describes how to install the Radia client to a remote computer using a GUI.

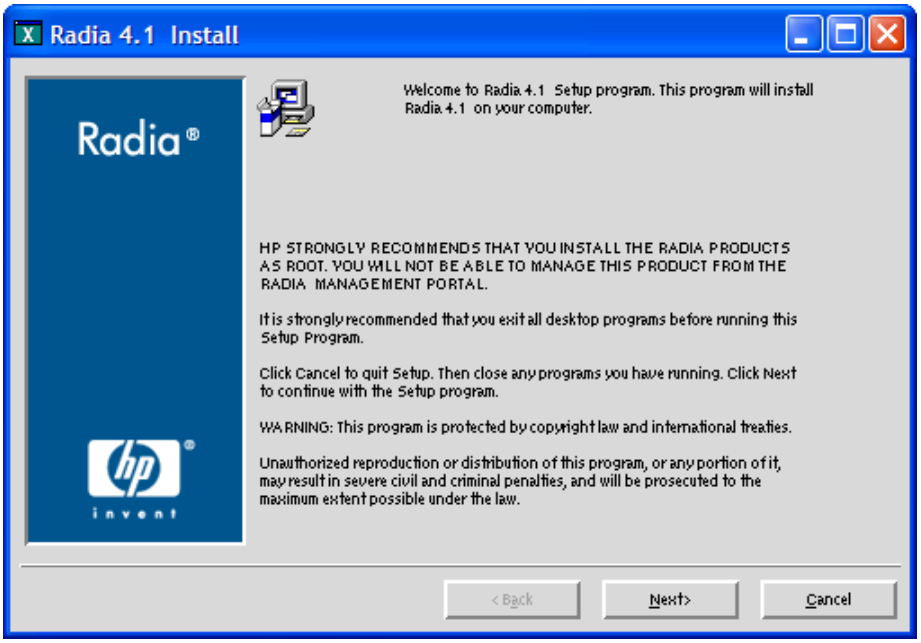
To install the Radia client to a remote computer using a GUI

1 Depending on your version of UNIX, change your current working directory to the correct `/client` subdirectory on the installation media.

Example: For HP-UX, type: **`cd /cdrom/hpux`**

2 Type **`./install`**, and then press **Enter**.

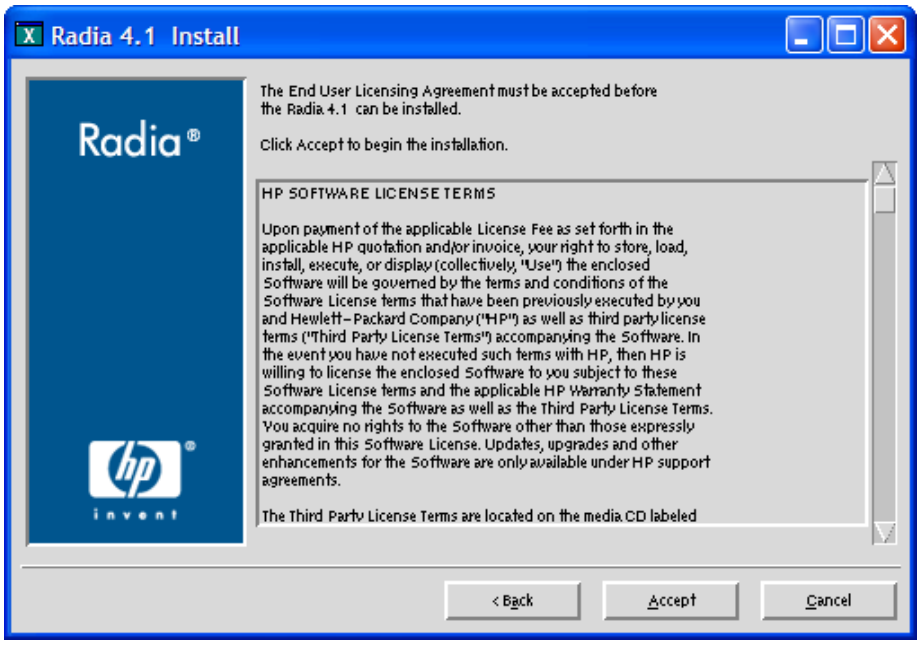
22 The Welcome window opens.



At any point during the installation, you can return to a previous window by clicking **Back**. Also, if you would like to exit the installation at any time, click **Cancel**.

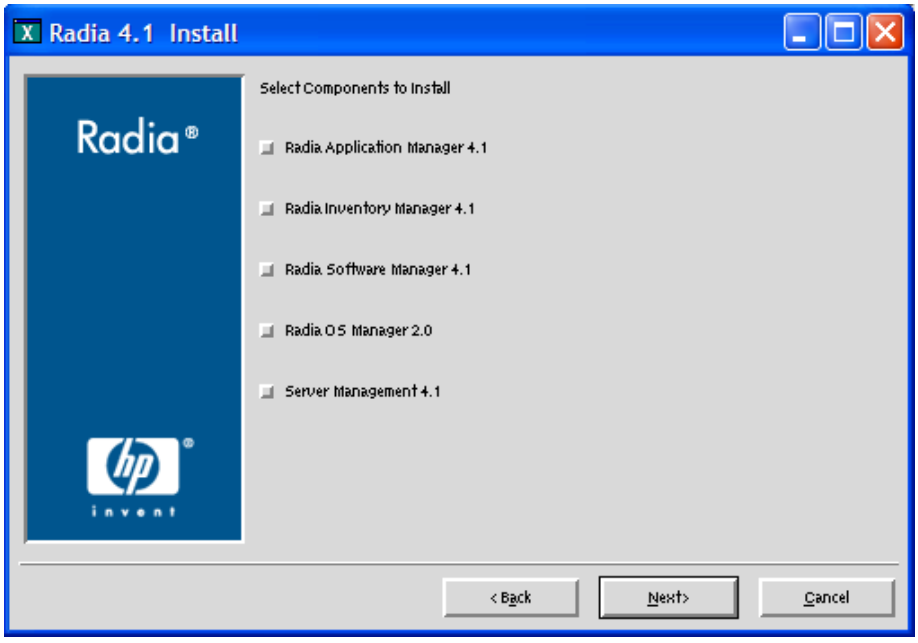
23 Click **Next**.

The End User License Agreement window opens.



24 Read the agreement and click **Accept** to continue.

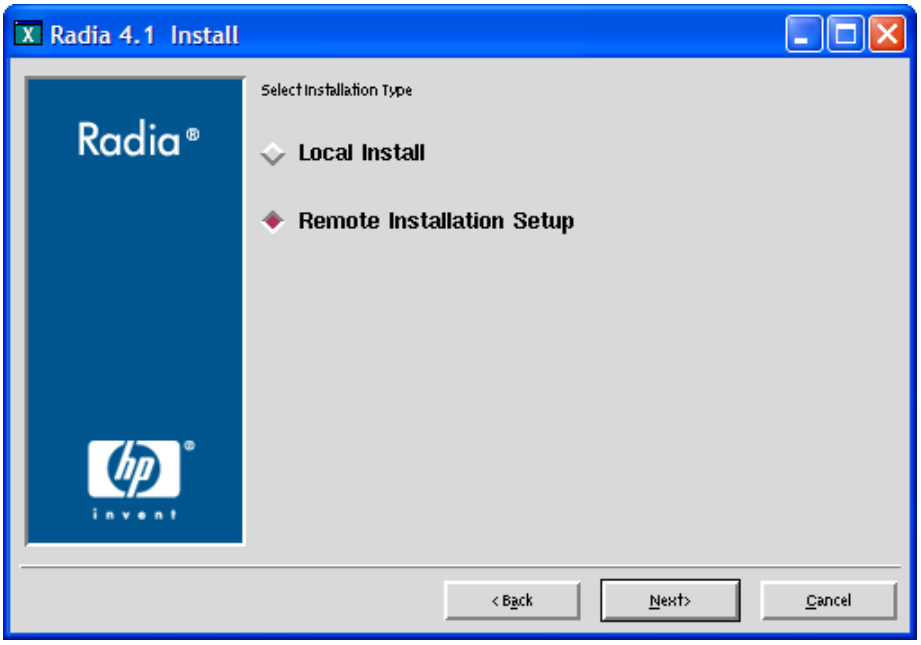
The Select Components to Install window opens.



25 Select the **Application Manager** check box.

26 Click **Next**.

The Select Installation Type window opens.

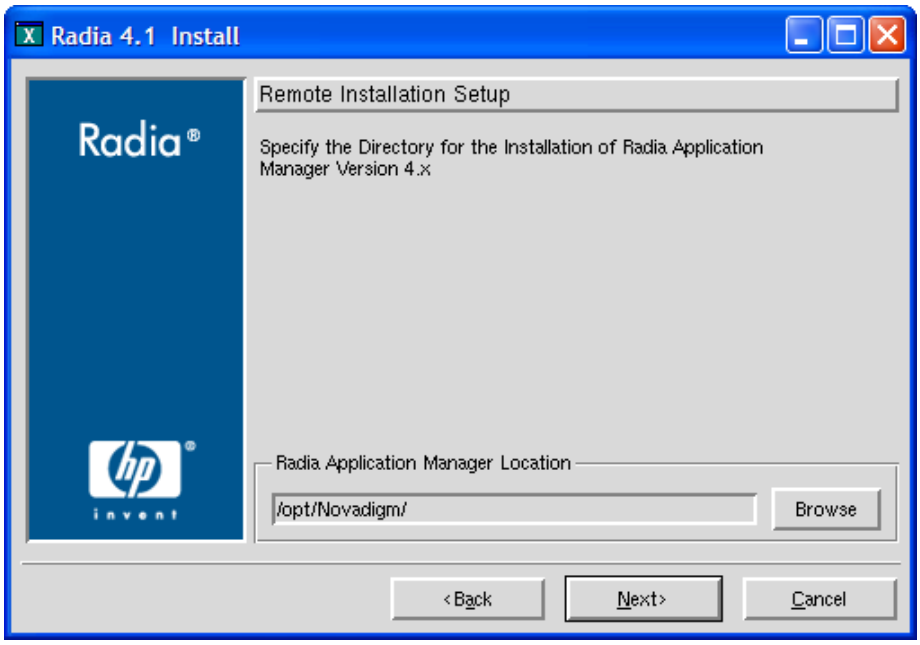


27 Select **Remote Installation Setup**.

This will store the installation media to another location on disk to be used later as the source for other client installations.

28 Then click **Next**.

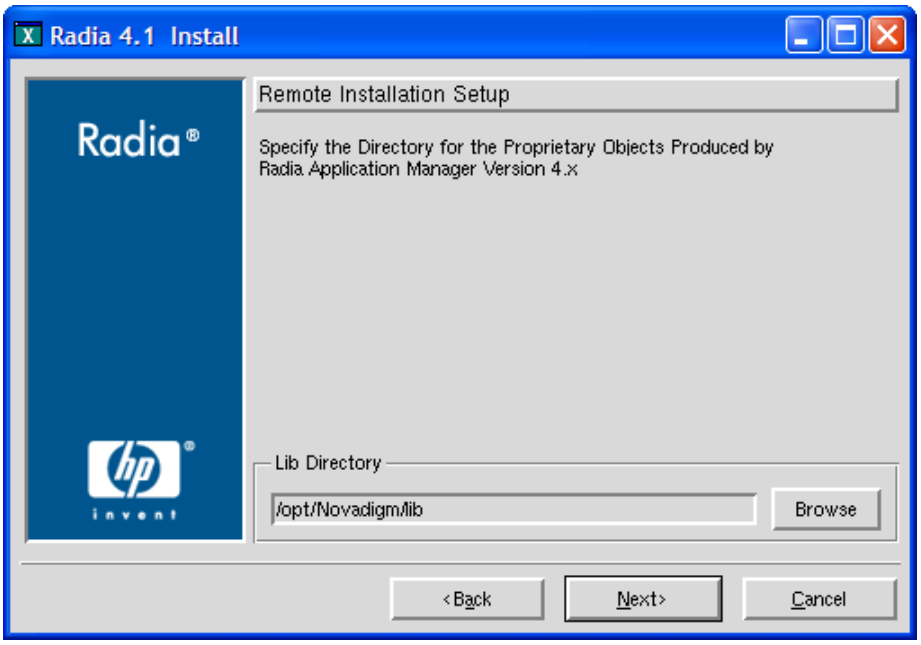
The Radia Client Location window opens.



29 Type the name of the directory where you want to install the Radia client executables for a silent installation, or click **Browse** to navigate to it.

30 Click **Next**.

The Lib Directory window opens.



31 Type the name of the directory where you would like to store proprietary information created by Radia for a silent installation, or click **Browse** to navigate to it.

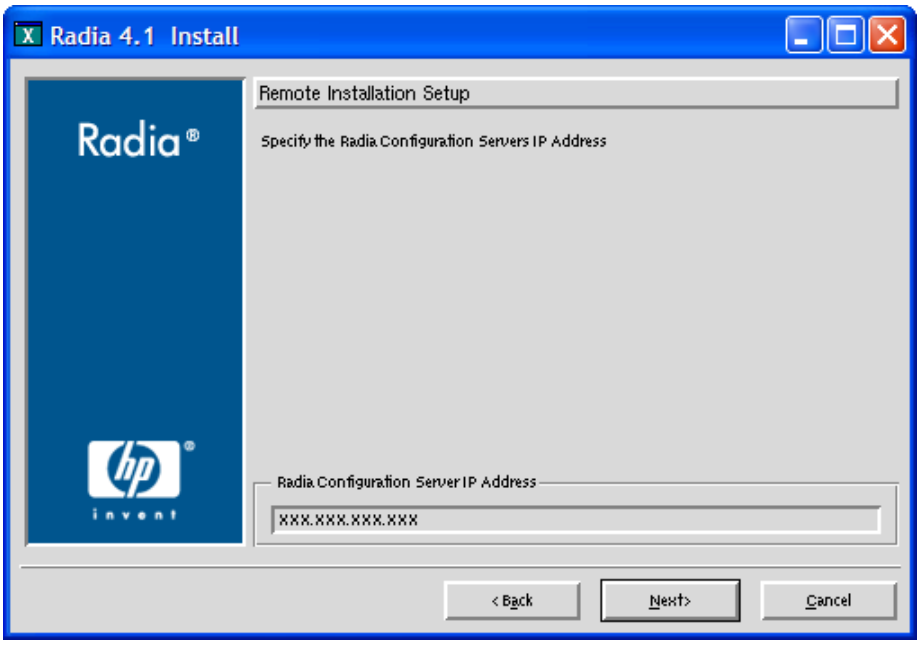
32 Click **Next**.

The Log Directory window opens.

33 Type the name of the directory where you would like to store log files generated by Radia for a silent installation, or click **Browse** to navigate to it.

34 Click **Next**.

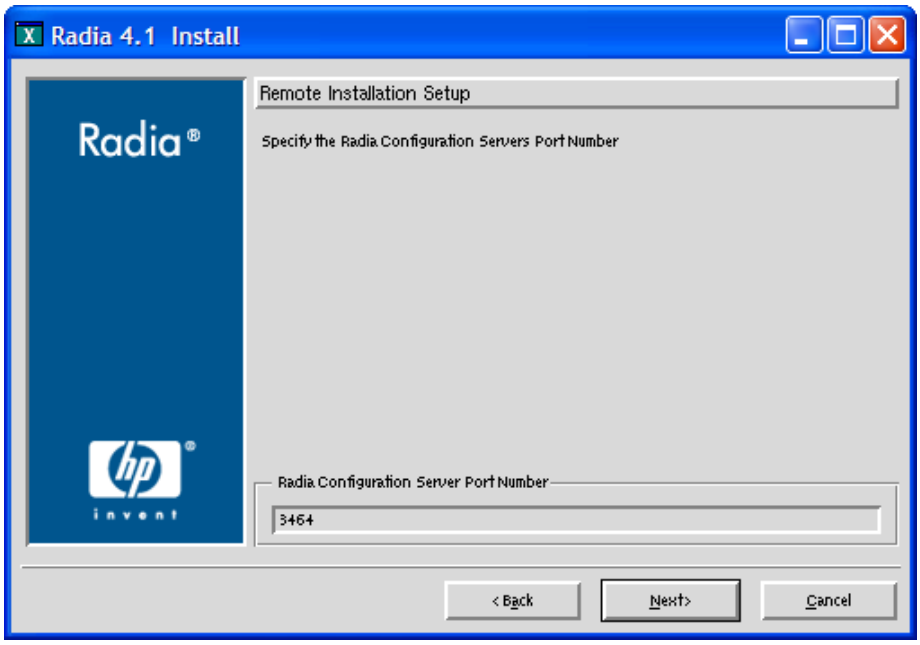
The Configuration Server IP Address window opens.



35 Type the IP address (format: xxx.xxx.xxx.xxx) of the Configuration Server that the Radia client will connect to. Specify a valid IP address or hostname recognized by the client workstation.

36 Click **Next**.

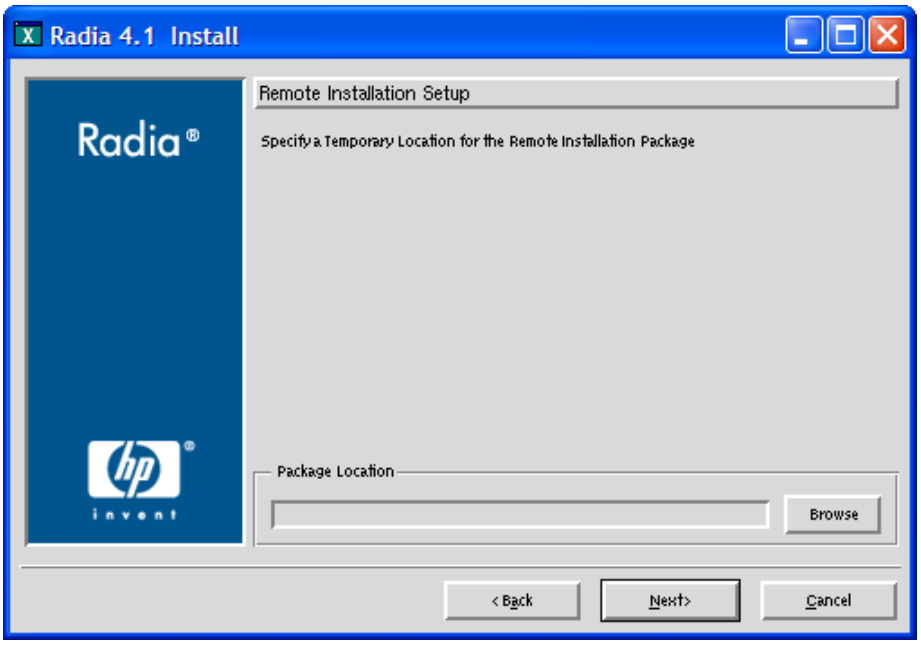
The Configuration Server Port Number window opens.



37 Type the port number of the Configuration Server (default is 3464).

38 Click **Next**.

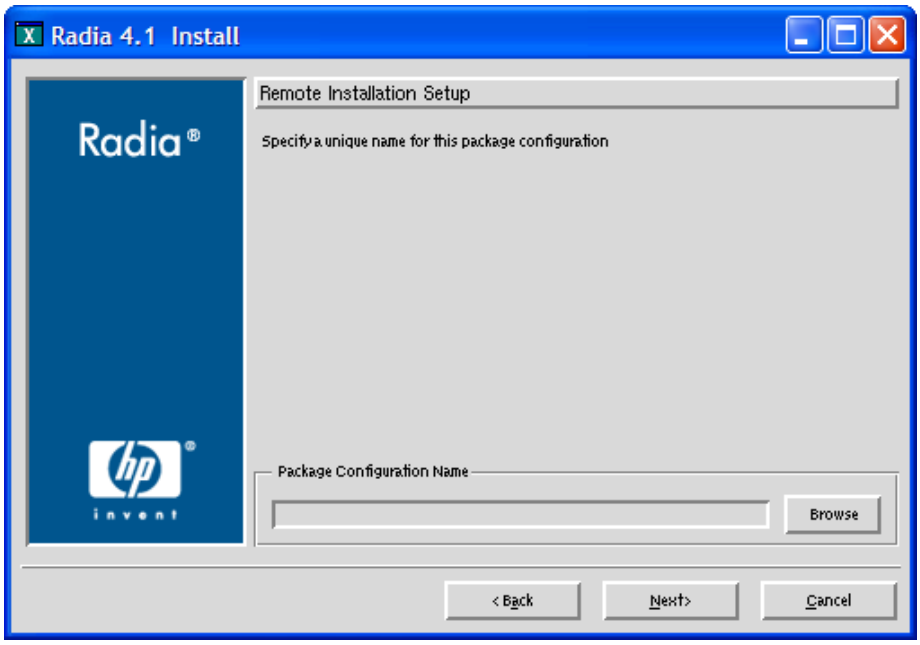
The Package Location window opens.



39 Type the fully qualified path to a directory where you would like to store the Radia client installation media for future client installations, or click **Browse** to navigate to it.

40 Click **Next**.

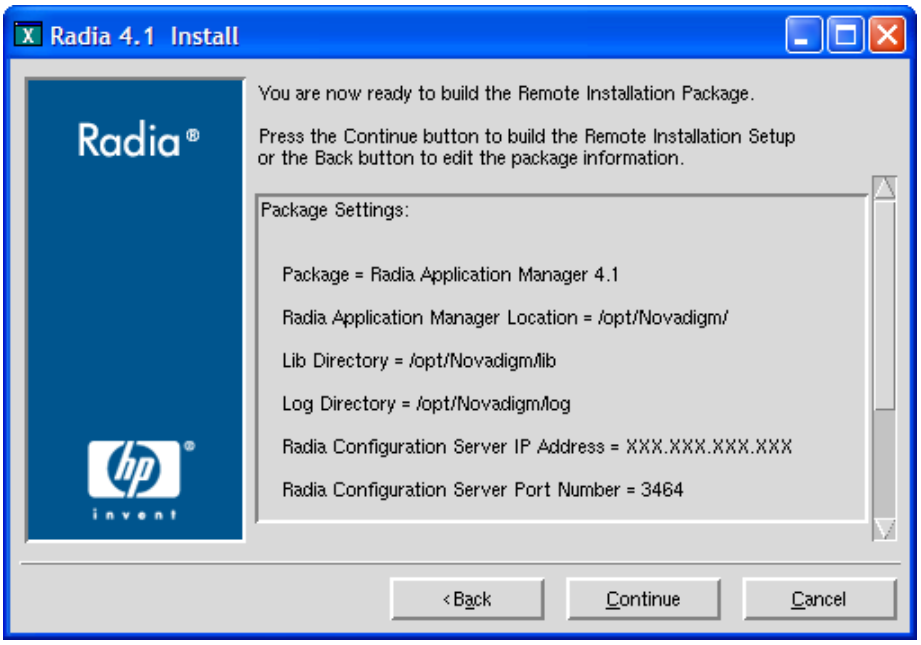
The Package Configuration Name window opens.



41 Type the fully qualified path to a configuration file that you would like to use for silent installations, or click **Browse** to navigate to it. The configuration file you specify will contain the installation information you chose during the Remote Installation Setup.

42 Click **Next**.

The Package Settings window opens.



43 Review the settings displayed in the Package Settings window.

44 Click **Continue** to build the Remote Installation Package.

45 When complete, click **Finish** to exit the installation.

The Radia client installation media has been successfully stored on disk for future installations.

Once the media has been stored for other computers to use for remote installations, you should become familiar with the variables in the configuration file.

Customizing the Installation Configuration File

A configuration file supplies the default responses for silent Radia client installations. These responses would normally be provided during an interactive Radia client installation. When performing silent installations, additional installation options are also available in the configuration file.

The variables available in the configuration file are described in Table 4, on page 56.

Table 4: Configuration File Variables

Variable	Sample Value	Description
REMOTE	0	0 designates a local installation. 1 designates a Remote Installation Setup.
INSTDIR	/opt/Novadigm	The default installation directory.
IDMLOG	/opt/Novadigm/log	This can be defined to designate a directory for IDMLOG other than the default INSTDIR/log.
IDMLIB	/opt/Novadigm/lib	This can be defined to designate a directory or IDMLIB other than the default INSTDIR/lib.
PREPROC		The fully qualified name of a script or executable to run pre-installation.
PREPARM		Any parameters that may be required by the pre-installation method specified in the variable PREPROC.
POSTPROC		The fully qualified name of a script or executable to be run post-installation.
POSTPARM		Any parameters required by the post-installation method specified in the variable POSTPROC.
MGRIP	1.1.1.98	The default IP address for connection to the Configuration Server.
MGRPORT	3464	The default port number for connection to the Configuration Server.
NTFYPORT	3465	The default Notify port used.

Variable	Sample Value	Description
CONNECT	Y	Connects to the Configuration Server immediately after the installation. Default behavior is N. Set to Y if you want your Radia client to connect to the Configuration Server automatically after the installation.
OBJECTS	./object.txt	The file that is used to create or update Radia attributes after the installation.
DUAL	1	0 designates RAM only selected. 1 designates more than one component selected.

Using a Pre- or Post-Installation Script

You can create and run custom executables or shell scripts prior to or after the silent installation of a Radia client. For example, your post-installation script can initiate a connection to the Configuration Server in order to process mandatory applications. The example below is part of a shell script that initiates the connection to the Configuration Server and processes mandatory applications.

```
#!/bin/sh
#
cd /opt/Novadigm

# ZIPADDR is the IP address or hostname of the manager
ZIPADDR="xxx.xxx.xxx.xxx"
# ZDSTSOCK is the TCP port the manager is running on
ZDSTSOCK="3464"

# To manage the machine
# 1. .edmprof must exist in root's home directory
# 2. The connect must be run as root

/opt/Novadigm/radskman
mname=NVDM,dname=SOFTWARE,ip=$ZIPADDR,port=$ZDSTSOCK,cat=promp
t,ind=y,uid=\$MACHINE,startdir=SYSTEM,ulogon=n
```

Customizing Installed Object Variable Content

The configuration file option OBJECTS allows you to specify the fully qualified path to a filename that contains data in the form:

```
OBJECT_NAME VARIABLE_NAME VARIABLE_VALUE
```

```
An example of a valid object file is:
```

```
ZMASTER ZTRACE N
```

```
ZMASTER ZTRACEL 000
```

When creating an object text file:

- A pound sign (#) at the beginning of a line indicates a comment.
- A pound sign (#) on any other part of a line will be considered data.
- The format is OBJECT_NAME followed by VARIABLE_NAME. Everything after the VARIABLE_NAME is considered VARIABLE_VALUE.
- The VARIABLE_VALUE text should not be enclosed by any special characters.

Performing a Silent Installation of a Radia Client

We recommend the client be installed as root.

Performing a silent installation of the Radia client using stored Radia client installation media requires that:

- your Radia system administrator has already run the Remote Installation Setup installation method.
- the workstation running the silent installation is able to access the directory path where the installation media was stored.

Several parameters can be used on the command line when performing a silent installation of the Radia client. Table 5 on page 59 describes these.

Table 5: Silent Installation Command Line Parameters

Parameter	Example	Description
`-cfg`	`./install -cfg Radia.cfg`	The filename specified after `-cfg` is the name of the configuration file to be used during the installation. For information about configuration files, see *Customizing the Installation Configuration File* on page 55.
`-mode silent`	`./install -mode silent -cfg /common/Radia.cfg`	Installs the Radia client in silent mode based on the parameters set in the configuration file specified after the `-cfg` parameter. For information about configuration files, see *Customizing the Installation Configuration File* on page 55.

Non-graphical Installation

This section describes a non-graphical (using a command line) installation of the Radia client for UNIX.

To install the Radia client for UNIX using a command line

These instructions guide you through the local non-graphical installation of the Radia client for UNIX. For the graphical installation, see Graphical Installation on page 34.

1 Depending on your version of UNIX, change your current working directory to the correct subdirectory on the installation media.

Example: **For HP-UX, type:** **`cd /cdrom/hpux`**

2 Type **`./install -mode text`**, and then press **Enter**.

The Radia client installation begins.

46 Type **C**, and press **Enter**.

47 Press **Enter** to accept the default component, the Application Manager.

48 Press **Enter** to skip the installation of the Inventory Manager.

OR

Type **Y** to install the Inventory Manager.

49 Press **Enter** to skip the installation of the Software Manager.

OR

Type Y to install the Software Manager.

50 Select the type of installation. The default is 1, a local installation.

Type **1**, and then press **Enter** to install the Radia client locally.

OR

Type **2**, and then press **Enter** to set up remote installation media.

For this example, we accepted the default.

51 Specify the installation location for the Radia client, and then press **Enter**.

52 Specify the location for the Radia proprietary objects (IDMLIB), and then press **Enter**.

53 Specify the location for the log files created by Radia (IDMLOG), and then press **Enter**.

54 Specify the IP address of the Configuration Server, and then press **Enter**.

55 Specify the port number for the Configuration Server, and then press **Enter**.

56 Review the installation settings you've chosen.

57 If you would like to install the Radia client with these parameters, press **Enter** to accept the default answer of **Y**.

If you want to change any of these settings, type **N** to re-enter the installation information.

58 When you're satisfied with the settings, press **Enter** to install the Radia client.

The Radia client is installed.

About Radia Daemons in UNIX

The Radia client installation program installs the following daemon executables:

- **Radia Notify (default port 3465)**
  Use Radia Notify, **radexecd**, to push updates to subscribers or to remove applications. A Notify message is sent from the Configuration Server to this daemon. When the daemon receives the Notify message, the Application Manager connects to the Configuration Server and performs the action initiated by the Notify operation.

  If you want to send a Notify to subscribers of a particular application, that application *must* be installed on their computers in order for them to be eligible for notification.

- **Radia Scheduler**
  Use the Radia Scheduler service, **radsched**, to schedule timer-based deployments of applications.

The installation of **radexecd** and **radsched** as services on a UNIX workstation is not automated within the context of the installation. The starting of services on UNIX workstations is operating system dependent. For information about installing Radia daemons as system services at boot time, see your local UNIX system administrator or refer to your UNIX operating system's manual.

Sample Shell Scripts

The installation of the Radia client includes a subdirectory called "sample". It contains a sample shell script called **daemons.sh** that may be used to start, stop, and restart the **radexecd** and **radsched** daemons.

- To start the radexecd and radsched daemons, type: `daemons.sh start`
- To stop the radexecd and radsched daemons, type: `daemons.sh stop`
- To stop, then restart the radexec and radsched daemons, type: `daemons.sh restart`

Troubleshooting the Client Installation

Should you encounter any problems while installing the Radia UNIX Client, please perform the following steps before contacting technical support:

- Enable diagnostic tracing by appending the text **`-loglevel 9`** to the installation command line and re-run the installation.
- Have this log file (`tmp/setup/setup.log`) located in the home directory of the UNIX user ID who ran the install.

The installation option `-loglevel 9` should only be used to diagnose installation problems.

Summary

- We strongly recommend that you install and run the Radia clients as root.
- Install the Radia clients using either the graphical or non-graphical modes.

3 Installing the Administrator Workstation for UNIX

At the end of this chapter, you will:

- Understand the system requirements and permissions necessary to install the Administrator Workstation for UNIX.
- Be able to install the Administrator Workstation using either the graphical or non-graphical mode.

If you are responsible for packaging applications or configuring them for distribution, install the Administrator Workstation on your administrator computer.

Use the Radia Packager to create software or data packages, and then promote them to the Radia Database.

System Requirements

- HP-UX Operating System Version 10.20 or above, PA Risc CPU.
- Red Hat Enterprise Linux Version 2.1 and 3.0, Intel Pentium processor or compatible CPU.
- SUSE LINUX versions 8 and 9, Intel Pentium processor or compatible CPU.
- Solaris Operating System Version 2.6 or above, SPARC CPU or Intel Pentium processor.
- AIX Operating System Version 4.3.1, 5L.
- TCP/IP connection to a computer running Configuration Server.
- Radia client requires 20 MB free disk space.

Prerequisites

- We strongly recommend installing the Radia clients as root.
- Install the Radia client on a local file system.
- If you intend to run any of the graphical components of the Administrator Workstation software, make sure the UNIX environment variable DISPLAY is set in your environment. If it is not, you will need to set this variable to indicate the hostname or IP address to which you would like to redirect the graphical display.

Table 6: Setting the DISPLAY Variable

In a…..	**Type….**
C shell	setenv DISPLAY IP address or hostname:0.0
Bourne, Bash, or Korn shell	DISPLAY=IP address or hostname:0.0 export DISPLAY

If the DISPLAY environment variable is not set in your environment, the installation will default to a non-graphical installation.

Troubleshooting

Should you encounter any problems while installing the Radia UNIX Client, please perform the following steps before contacting technical support:

- Enable diagnostic tracing by appending the text **-loglevel 9** to the installation command line and re-run the installation.
- Have this log file (`tmp/setup/setup.log`) located in the home directory of the UNIX user ID who ran the install.

The install option **–**`loglevel 9` should only be used to diagnose installation problems.

Recommendations

- We strongly recommend that you install and run the Administrator Workstation as root.

Root authority is required to apply owner and group designators to managed resources.

Installation Methods

You can install the Administrator Workstation by:

- Executing the installation procedure directly from the CD-ROM.
- Copying the files from the CD-ROM into a temporary directory and executing the installation procedure.

Several parameters can be used on the command line when installing the Administrator Workstation.

Table 7: Command Line Parameters

Parameter	Example	Description
-mode plain	./install –mode plain	Installs the Administrator Workstation in plain mode. The installation graphics are displayed in plain mode (no moving graphics). This is useful for remote installations where network bandwidth may be an issue.
-mode text	./install –mode text	Installs the Administrator Workstation in text mode using the non-graphical installation. The installation takes place completely on the command line. The installation will default to text mode if the DISPLAY environment variable is not set.

Installing the Administrator Workstation for UNIX

This section describes both the graphical (using a GUI) and non-graphical (using a command line) installations of the Administrator Workstation for UNIX.

Graphical Installation

This section describes how to install the Administrator Workstation for UNIX using a graphical user interface (GUI).

To install the Administrator Workstation for UNIX using a GUI

These instructions will guide you through the graphical installation of the Administrator Workstation. For non-graphical instructions, see Non-graphical Installation on page 76.

If the UNIX user ID of the person performing the Administrator Workstation installation has previously installed a Radia client, the location of the Administrator Workstation will default to the location of the Radia client executables.

1. Depending on your version of UNIX, change your current working directory to the correct subdirectory on the installation media.

   Example: For HP-UX, type: **`cd /cdrom/management infrastructure/administrator workstation/hpux/`**

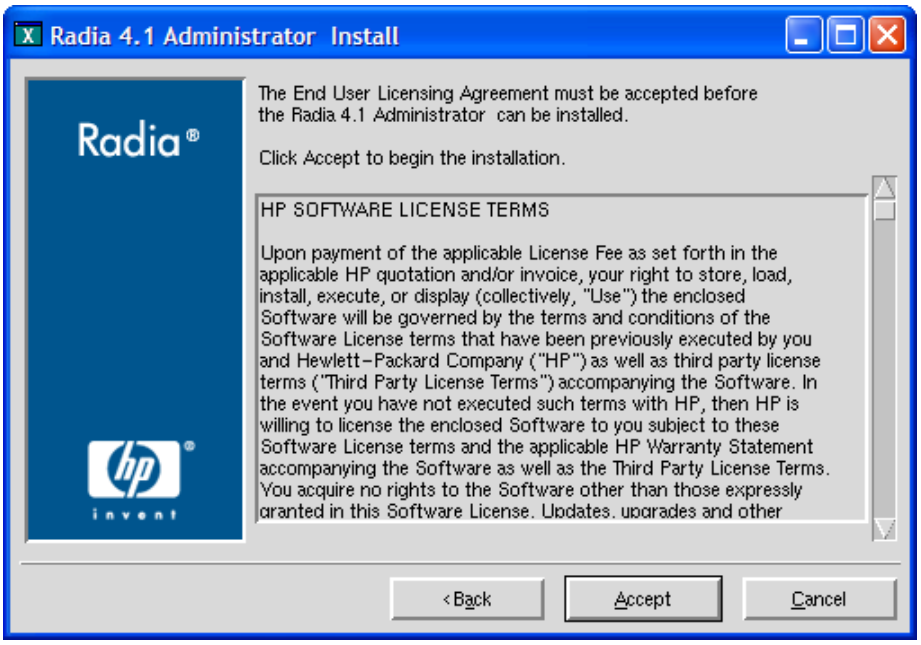
2. Type **`./install`**, and then press **Enter**.

   The Welcome window opens.

59 Click **Next**.

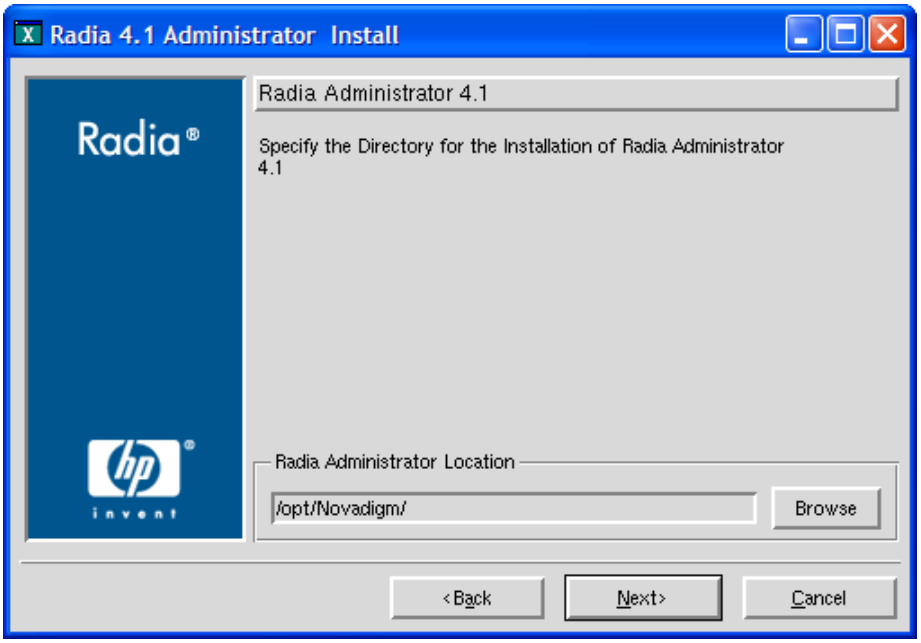
If you are installing the Administrator Workstation to a computer with the same UNIX user ID that had previously installed a Radia client, the installation program will prompt you for the Configuration Server's IP address next. The next three windows: Radia Administrator Location, Lib directory, and Log directory, are only needed if you are installing the Administrator Workstation to a computer that does not have a Radia client already installed.

The HP Software License terms window opens.



60 Read the software license terms and click **Accept**.

The Radia Administrator Location window opens.



61 Type the name of the directory where you are installing the Administrator Workstation, or click **Browse** to navigate to it.

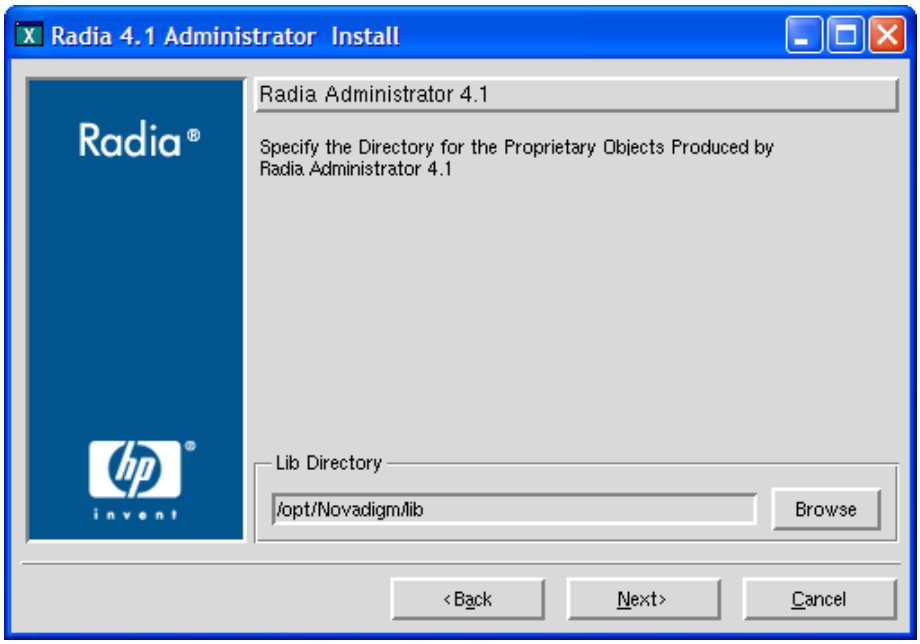
62 Click **Next**.

If the specified directory already exists you will be prompted to verify this location.

— If you would like to update the existing directory, click **OK**.

— If you would like to change the directory location, click **Cancel**.

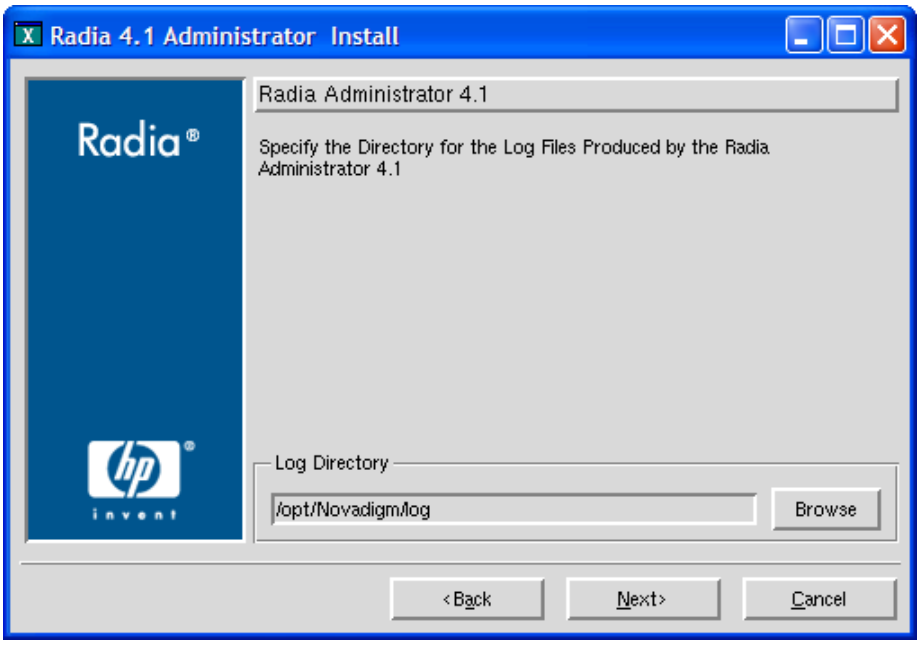
The Lib Directory window opens.



63 Type the name of the directory where you would like to store proprietary information created by Radia (the lib directory), or click **Browse** to navigate to it.

64 Click **Next**.

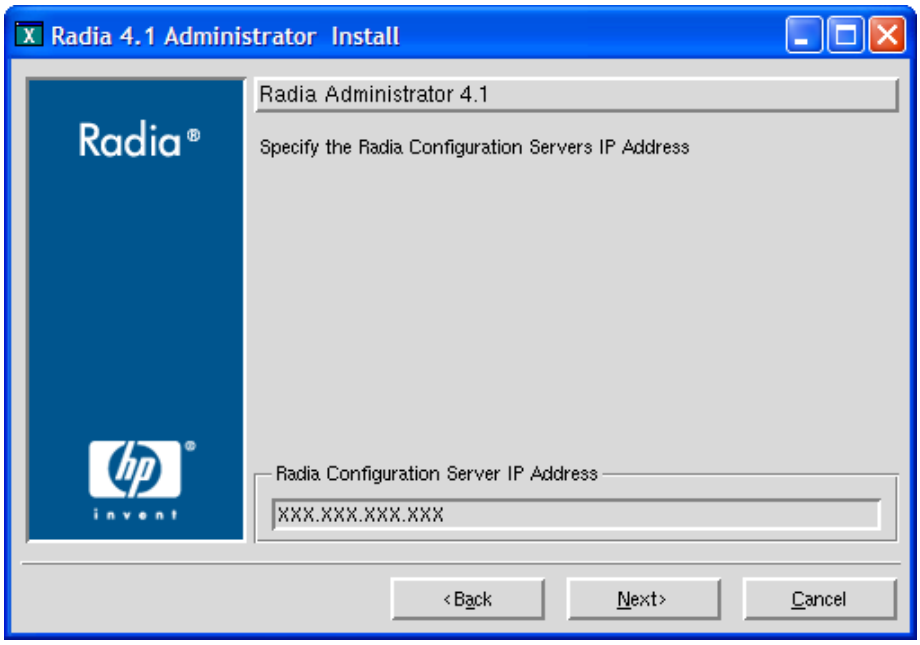
The Log Directory window opens.



65 Type the name of the directory where you would like to store the log files generated by Radia, or click **Browse** to navigate to it.

66 Click **Next**.

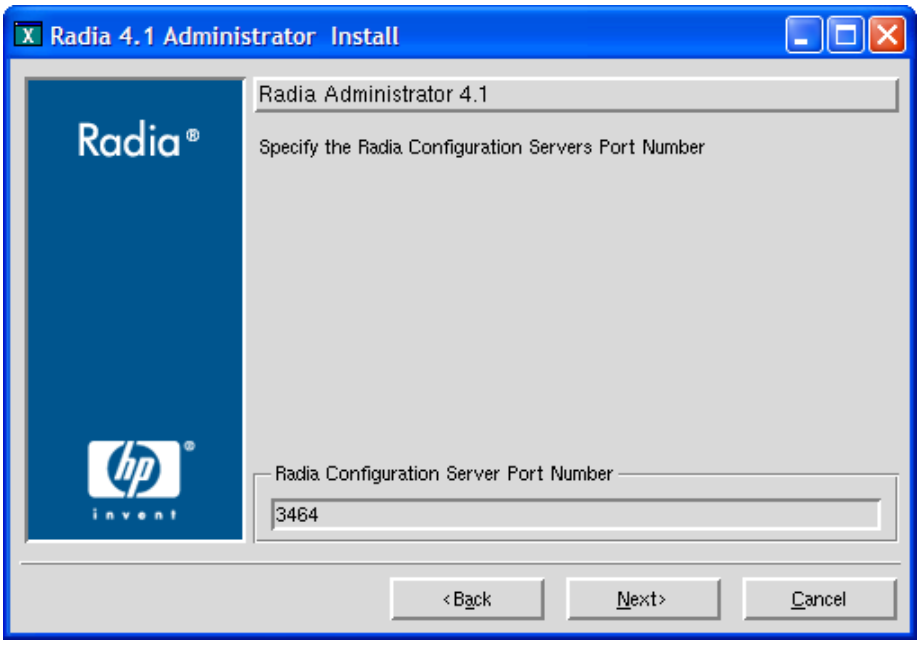
The Configuration Server IP Address window opens



67 Type the IP address (format: xxx.xxx.xxx.xxx) or hostname of the Configuration Server you will be publishing to.

68 Click **Next**.

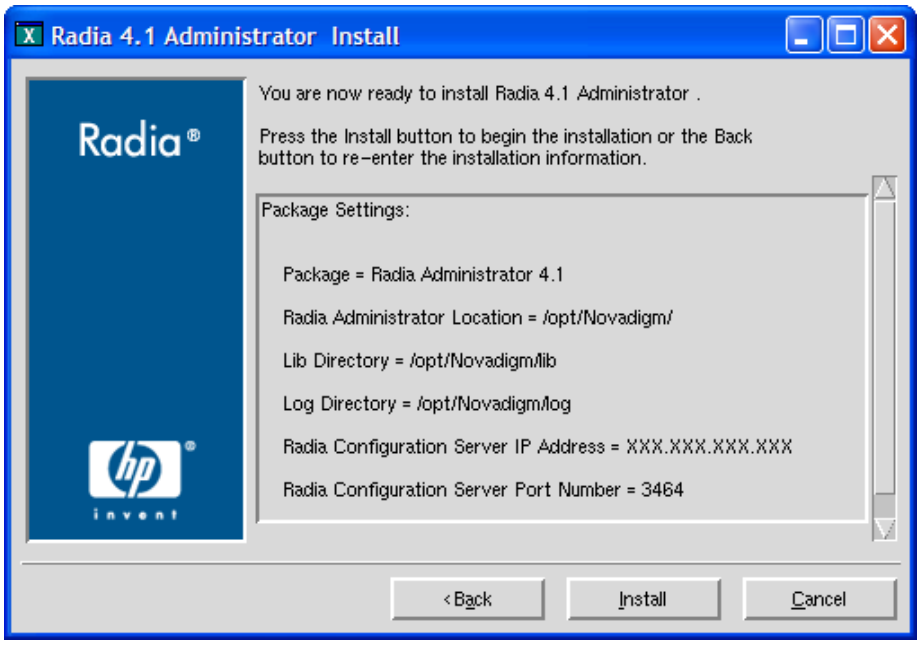
The Configuration Server Port Number window opens.



69 Type the port number of your Configuration Server (default is 3464).

70 Click **Next**.

The Package Settings window opens.



71 Review the settings displayed in the Package Settings window. If you would like to change any of the settings, click **Back** until you get to the appropriate window.

72 When you are satisfied with the Package Settings, click **Install**.

The Administrator Workstation is installed.

Non-graphical Installation

This section describes a non-graphical (using a command line) installation of the Administrator Workstation for UNIX.

To install the Administrator Workstation for UNIX using a command line

These instructions guide you through the non-graphical installation of the Administrator Workstation. For the graphical installation, see Graphical Installation on page 69.

Task 1 Depending on your version of UNIX, change your current working directory to the correct subdirectory on the installation media.

Example: For HP-UX, type: **`cd /cdrom/management infrastructure /administrator workstation/hpux/`**

1 Type **`./install –mode text`**, and then press **Enter**.

The Administrator Workstation installation begins.

2 Type **`C`**, and then press **Enter**.

If you are installing the Administrator Workstation to a computer with the same UNIX user ID that had previously installed a Radia client, the installation program will prompt you for the Configuration Server's IP address next. The next three sections: Radia Administrator Location, Lib directory, and Log directory, are only needed if you are installing the Administrator Workstation to a computer that does not have a Radia client already installed.

73 Specify the installation location for the Administrator Workstation, and then press **Enter**.

74 Specify the location for the Radia proprietary objects (IDMLIB), and then press **Enter**.

75 Specify the location for the log files created by Radia (IDMLOG), and then press **Enter**.

76 Specify the IP address of the Configuration Server, and then press Specify the port number of the Configuration Server, and then Review the installation settings you've chosen.

77 If you would like to install the Administrator Workstation with these settings, press **Enter** to accept the default (Y) and begin the installation or type **`N`**, to re-enter your installation information.

78 To complete the configured installation process, press **Enter**.

The Administrator Workstation is installed.

Summary

- We strongly recommend that you install and run the Administrator Workstation as root.
- Install the Administrator Workstation for UNIX using the graphical or non-graphical mode.

4 Installing the Administrator Workstation for Windows

At the end of this chapter, you will:

- Be able to install the Administrator Workstation for Windows.
- Understand the system requirements for installing the Administrator Workstation.
- Know how to install the Administrator Workstation using the Installation Wizard and command lines.
- Understand the feature settings of the Administrator Workstation.
- Know how to remove and repair the Administrator Workstation using the Installation Wizard and command lines.

The Administrator Workstation for Windows is required within a UNIX environment in order to access the System Explorer. Install the Administrator Workstation for Windows onto a 32-bit Windows platform.

The Administrator Workstation installation program uses Microsoft Windows Installer. The program consists of one MSI package with four feature sets—Radia Packager, System Explorer, Client Explorer, and 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 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 2000, Server 2003, or XP, you must have administrator rights to the computer to install the 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.

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 Administrator Workstation

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

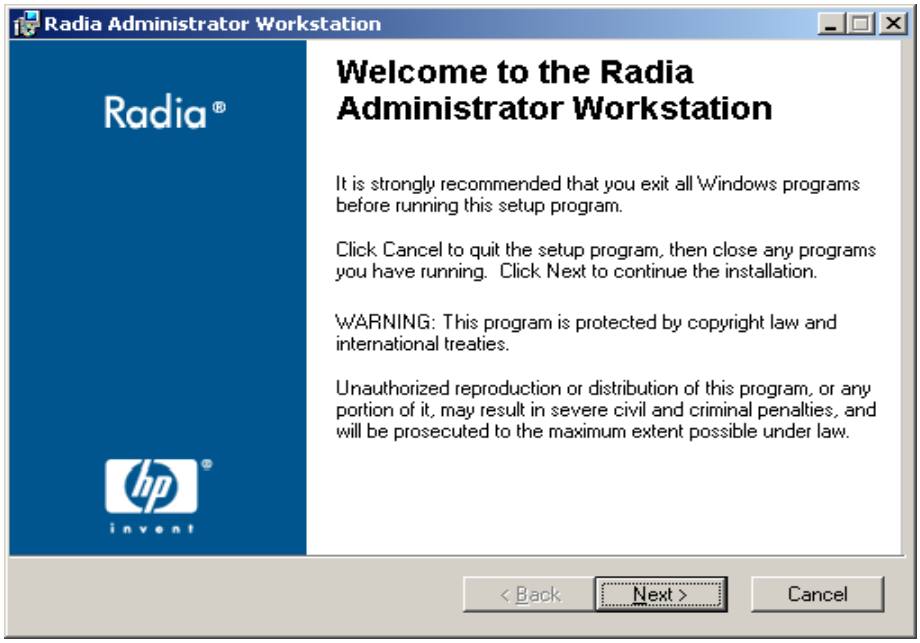
Using the Installation Wizard to Install the Administrator Workstation

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

To install the Administrator Workstation using the Installation Wizard

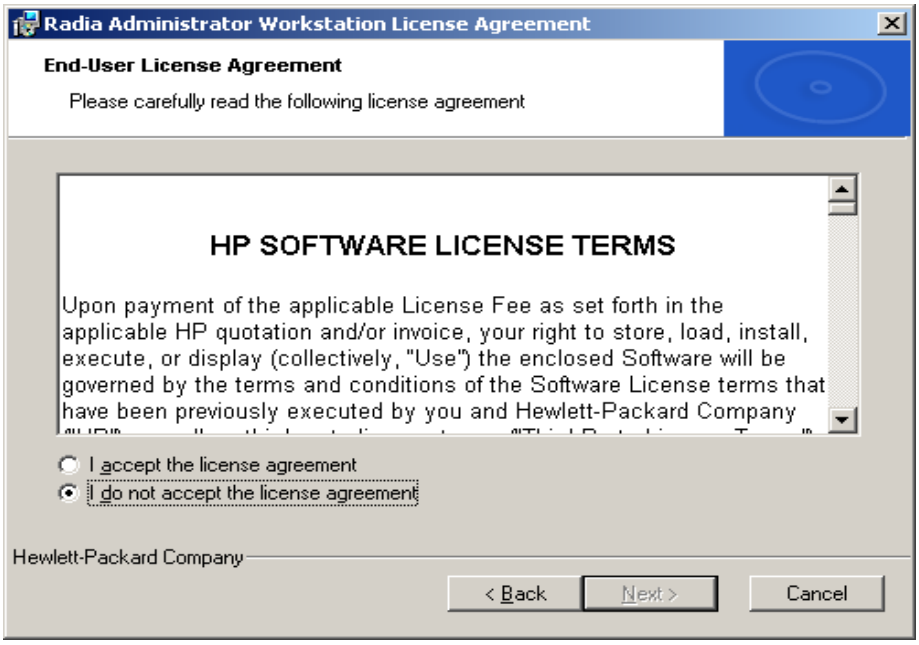
1 From the folder containing the Administrator Workstation installation files, run `setup.exe`.

The Administrator Workstation Installation Wizard opens.



2 Click **Next**.

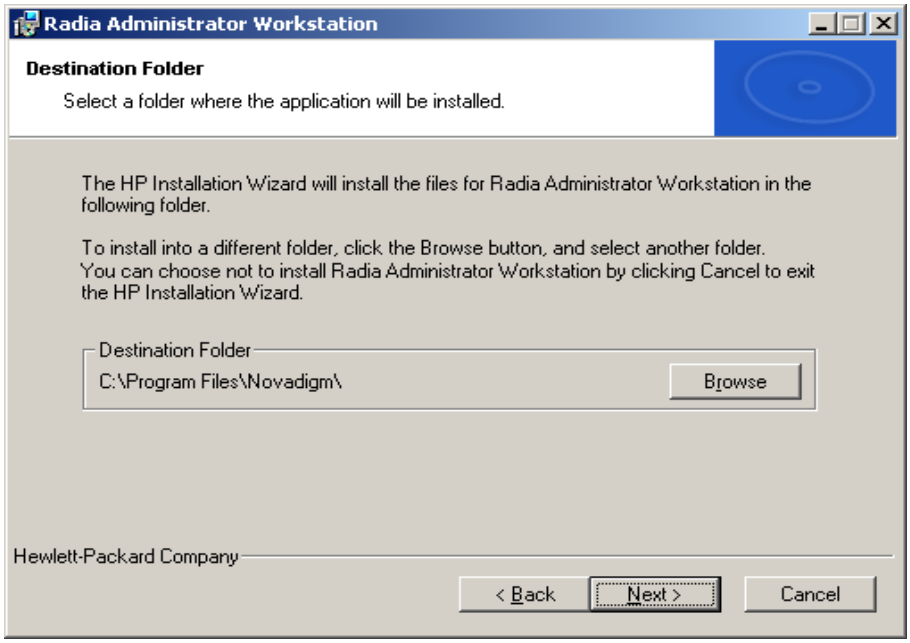
The License Agreement window opens.



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

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

If the Radia client is already installed on the computer, this window will not open and the Administrator Workstation is installed in the same location as the Radia client.

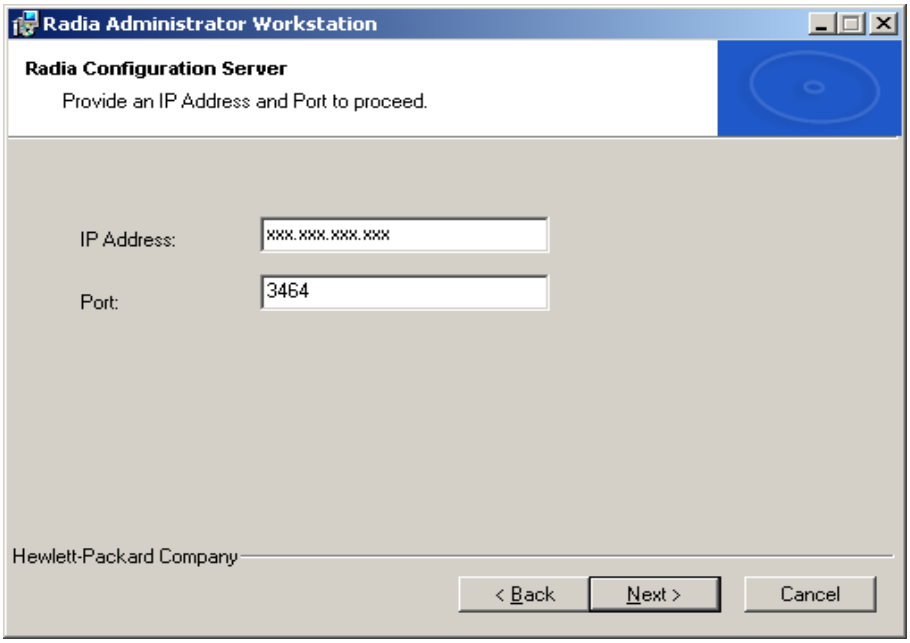


If you want to select a different destination for the Administrator Workstation, click **Browse**, and then navigate to the appropriate destination folder.

Click **OK** to continue.

80 Click **Next**.

The Configuration Server window opens.

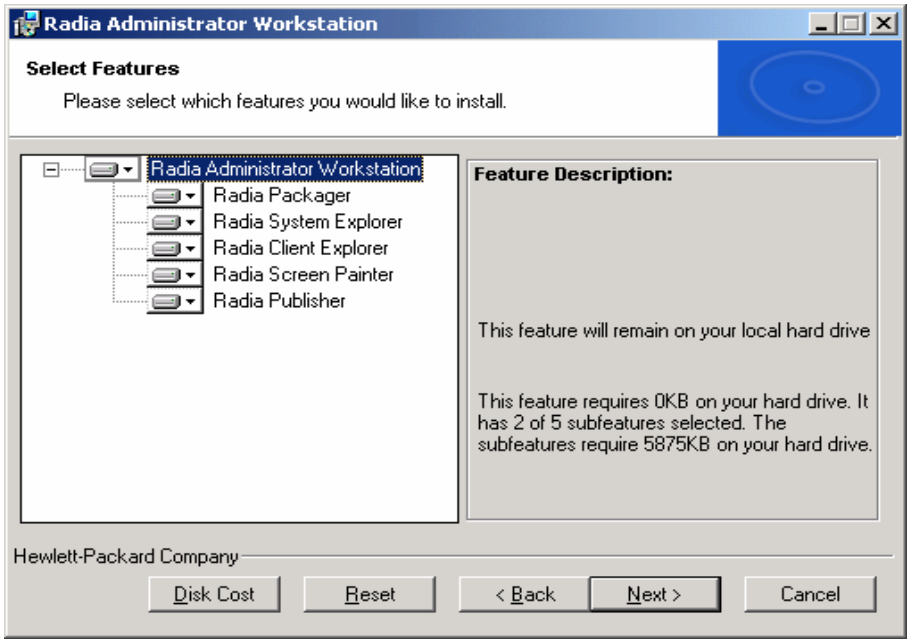


81 In the **IP Address** text box, type the IP address for the Configuration Server.

82 In the **Port** text box, type the port number (default is 3464).

83 Click **Next**.

The Select Features window opens.

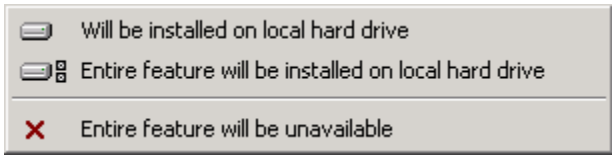


84 Click  to select the features that you would like to install.

If you want to set the same options for all of the features, you can click  next to 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  a shortcut menu for that feature opens.



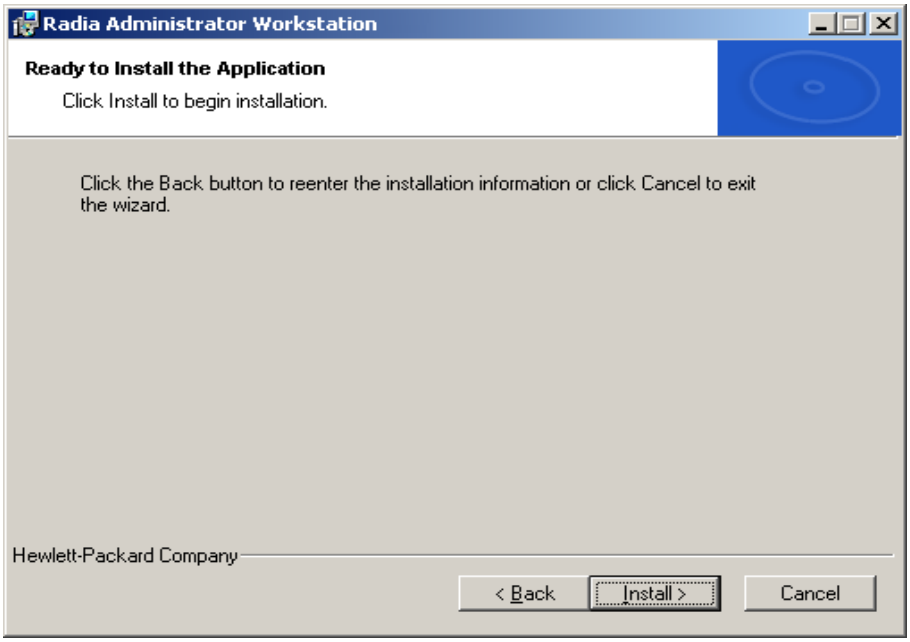
85 From the shortcut menu, select an installation option. These options are described in Table 8 on page 87.

Table 8: Feature Settings for the 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.

86 Click **Next**.

The Ready to Install the Application window opens.



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

88 When the installation is complete, click **Finish** to exit the program.

Using a Command Line to Install the Administrator Workstation

You can also use the command line to run the 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 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 9: 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 Administrator Workstation features as follows:

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

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

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

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

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

Additional Command Line Arguments

Some additional arguments that you can pass to the installation program on the command line are described in Table 10 below.

Table 10: 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.

Argument	Description
`/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 Administrator Workstation

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

Using the Installation Wizard to Remove the Administrator Workstation

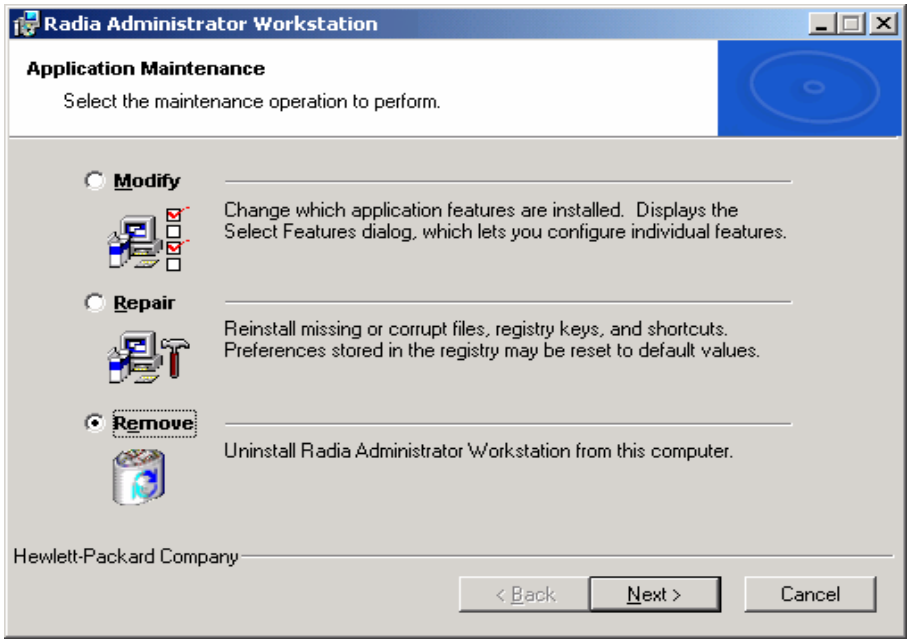
This section describes how to remove (uninstall) the Administrator Workstation using the Installation Wizard.

To remove specific features of the Administrator Workstation, use the Modify option on the Application Maintenance window. This is discussed in Modifying the Administrator Workstation Installation on page 100.

To remove the Administrator Workstation using the Installation Wizard

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

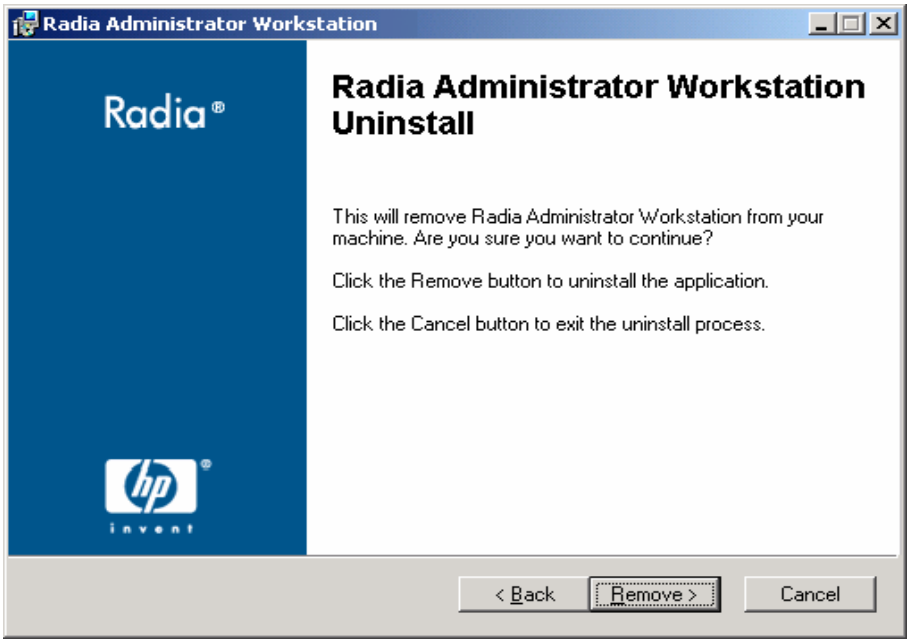
The Application Maintenance window opens.



2 Select the **Remove** option.

89 Click **Next**.

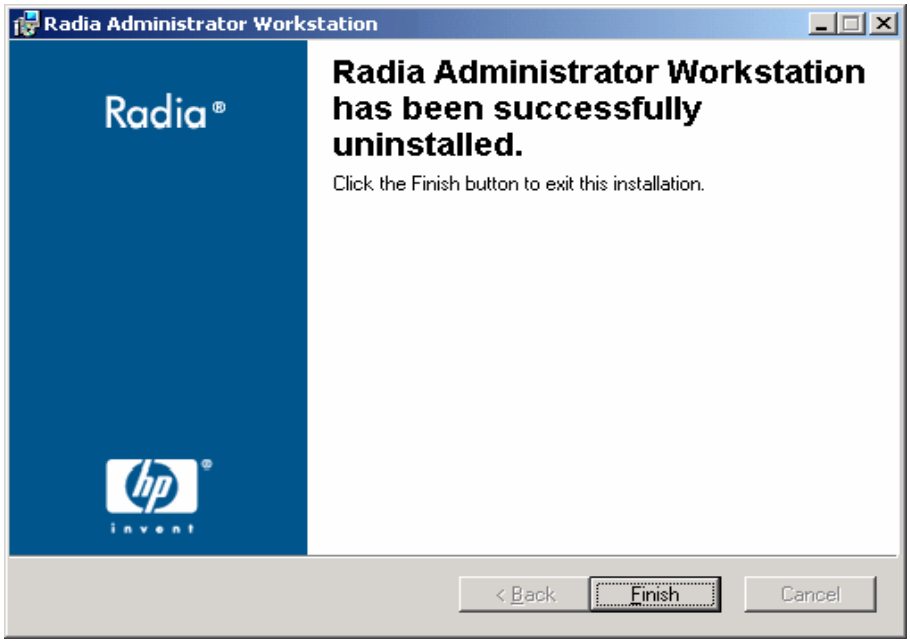
The Administrator Workstation Uninstall window opens.



90 Click **Next**.

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

The Administrator Workstation has been successfully uninstalled window opens.



91 Click **Finish**.

Using a Command Line to Remove the Administrator Workstation

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

To remove the Administrator Workstation using a command line

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

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

  OR

  If you would like to remove a single 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 System Explorer and Radia Client Explorer, type:

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

Reference the features for the Radia 4.x Administrator Workstations as follows:

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

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 Administrator Workstation

The Windows Installer installation program offers the ability to repair the Administrator Workstation. For example, if you have a missing 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 Administrator Workstation using the Installation Wizard and using a command line.

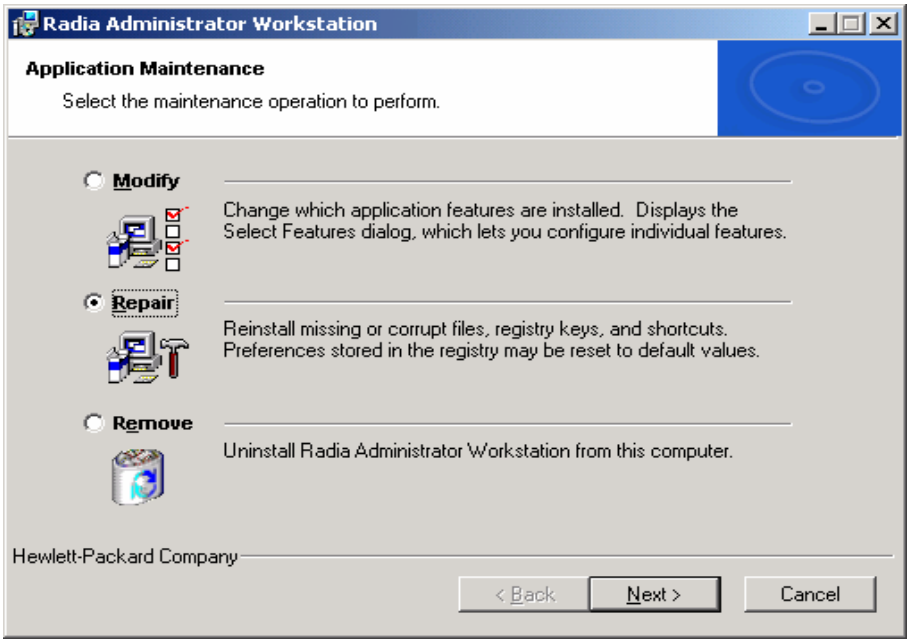
Using the Installation Wizard to Repair the Administrator Workstation

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

To repair the Administrator Workstation using the Installation Wizard

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

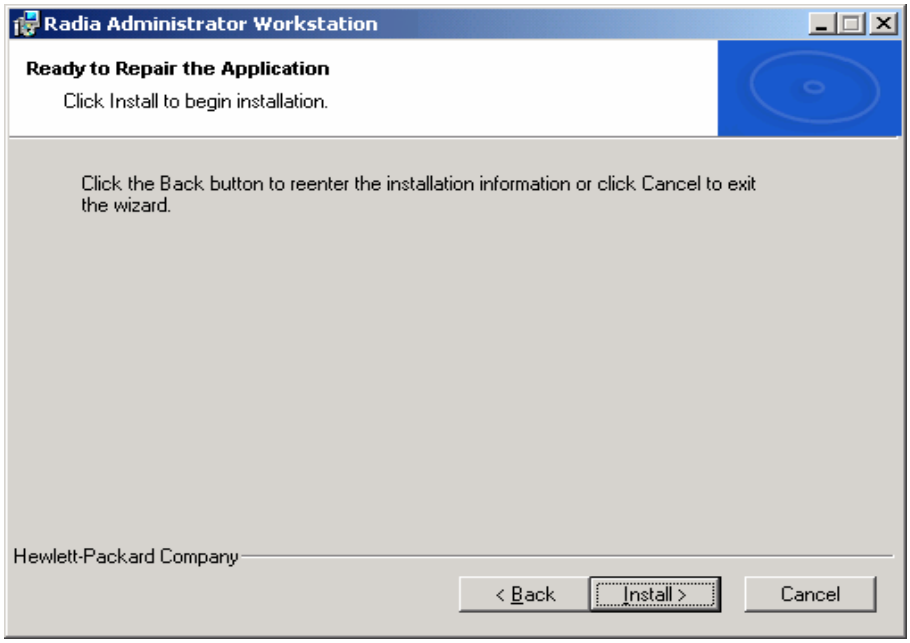
The Application Maintenance window opens.



2 Select the **Repair** option.

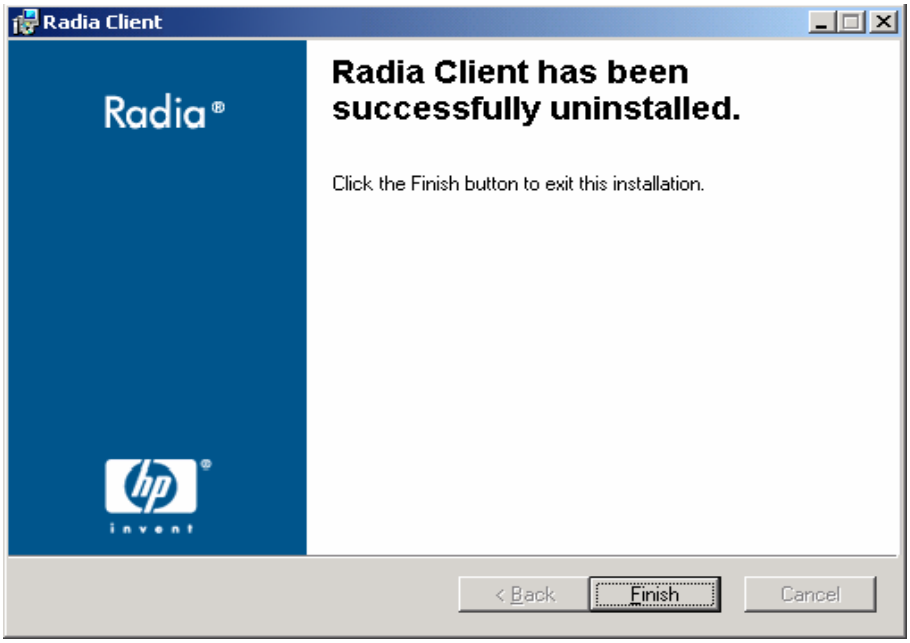
92 Click **Next**.

The Ready to Repair the Application window opens.



93 Click **Next**.

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



94 Click **Finish**.

Using a Command Line to Repair the Administrator Workstation

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

To repair the Administrator Workstation using a command line

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

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

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

Modifying the Administrator Workstation Installation

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

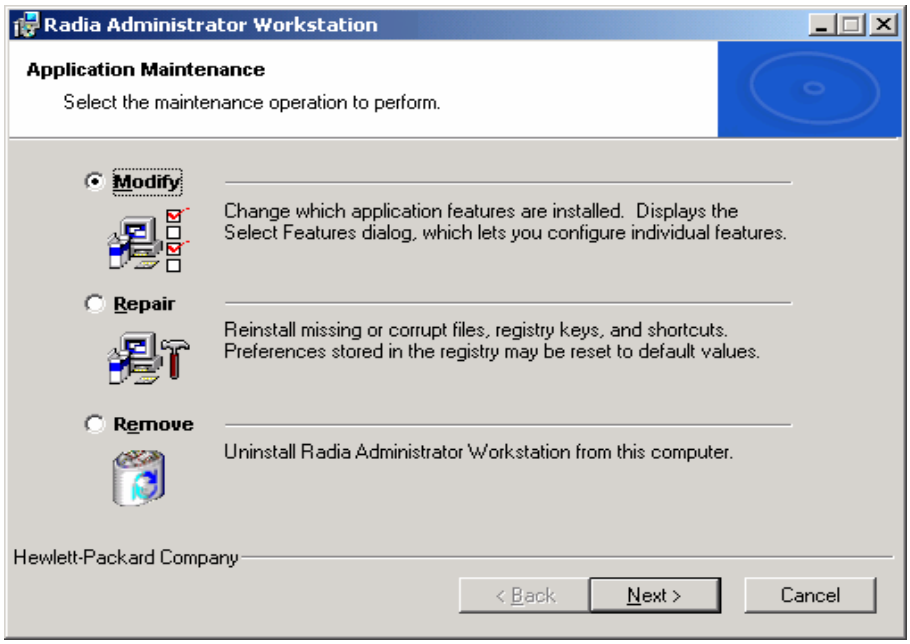
Using the Installation Wizard to Modify the Administrator Workstation

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

To modify the Administrator Workstation installation using the Installation Wizard

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

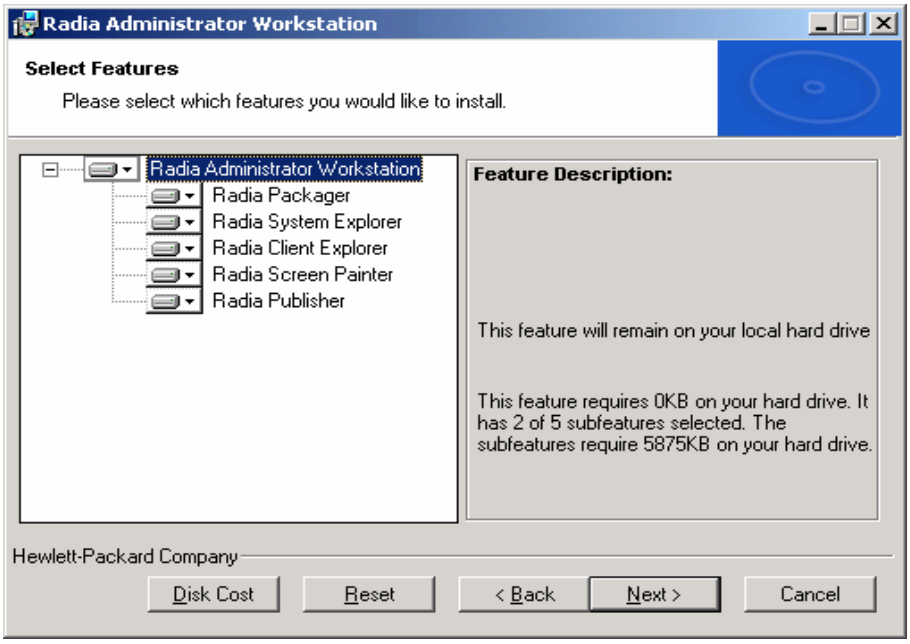
The Application Maintenance window opens.



2 Select the **Modify** option.

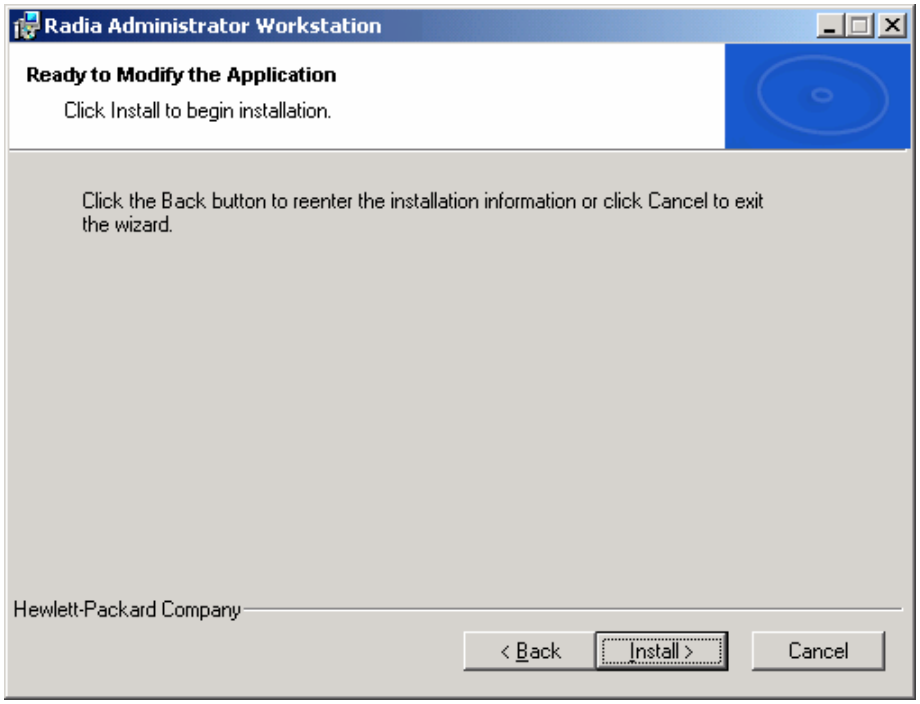
95 Click **Next**.

The Select Features window opens. See Installing the Administrator Workstation on page 82 for information about how to use this window.



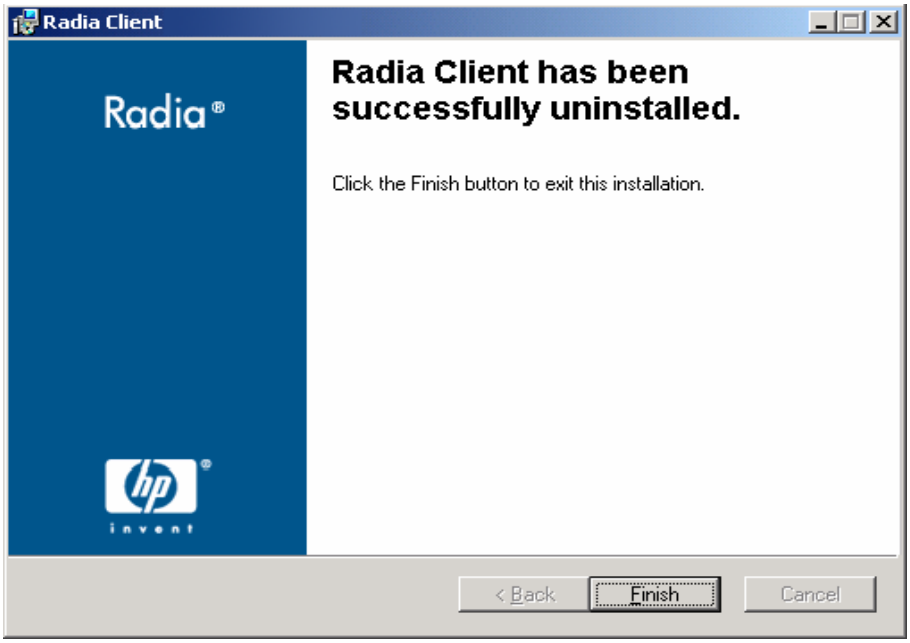
96 Click **Next**.

The Ready to Modify the Application window opens.



97 Click **Next**.

The Administrator Workstation has been successfully installed window opens.



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

Using a Command Line to Modify the Administrator Workstation Installation

To modify the Administrator Workstation installation using a command line

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

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

See Table 9 on page 88 for more information.

Example

If you want to install the Radia Packager to the local hard drive, and to make the 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 89 for additional arguments.

Summary

- The Administrator Workstation for Windows consists of one package with multiple feature sets.
- Install the Administrator Workstation on a clean computer.
- You can install the Administrator Workstation using a command line or using the Installation Wizard.

5 Packaging Applications and Content

At the end of this chapter, you will:

- Understand the packaging process.
- Understand the requirements for publishing software or content.
- Be able to publish an application using Component Selection Mode
- Be aware of the Publishing Adapter.
- Use the New Application Wizard in the System Explorer to create a service.
- Be able to prepare and distribute maintenance packages to the Application Manager.

This guide covers the *suggested* implementation for the 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 Application Manager. This chapter focuses on packaging.

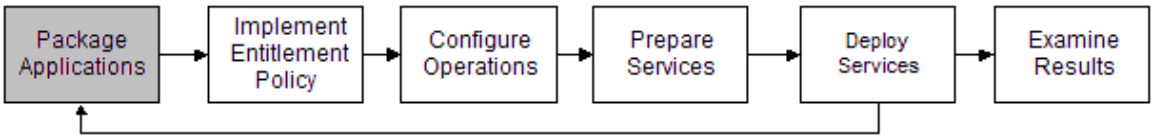


Figure 3: Tasks completed in this guide.

The Radia Publisher for Radia Version 3.x has been renamed the Radia Packager for Radia Version 4.x.

About Radia Packager

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 or package. 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 (a.k.a. the Configuration Server Database). A package must be published before its content can be distributed and deployed into your environment.

For the UNIX version of the Radia Packager, there is one publishing mode available, **Component Selection Mode.** In Component Selection Mode, you select the individual components that make up the application, such as files, directories, and links.

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 **UNIX File Resources (UNIXFILE)** instance for each file in the package.

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

Above are some of the default classes available in the SOFTWARE domain. You can also add your own classes to the Radia Database.

Then, you will use the System Explorer to create a service, assign policies, and prepare the package for deployment. See Chapter 6, Implementing Entitlement Policy and Chapter 9, Deploying Services for more information.

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Packaging Considerations Checklist

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

General

- ❑ 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 Setting the Required System Configuration on page 131 for more information.

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

Activation Options

- When do you want to activate the application—immediately on distribution or at a later time?

  See Setting Date and Time Constraints on page 132 for more information.

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

  See Chapter 9, Deploying Services for more information.

- Do you want to build and maintain versions?

  See Chapter 9, Deploying Services for more information.

Data Options

See Data Options Tab on page 140 for more information.

- What type of compression do you want to use?
- Are you distributing maintenance to the Application Manager client?
- How do you want to promote the resources?
- Are you sending out an update and only want to deploy the changes?

Verify Options

See Client Management Tab on page 136 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?

Delivery Options

See Client Management Tab on page 136 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 with the Application Manager.
- ❑ Do you want the data deployed under the user or machine context?

Client Behaviors

See Client Behaviors Tab on page 141 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?

Setting Default Properties

Soon, you will learn how to create packages using the Radia Packager. However, before creating a package, you may want to set default properties (such as compression settings and verification options) for the files and directories that will be included in the package. Changes made to the default properties also apply to new Radia Packager sessions. If necessary, you can modify the properties later for any file or directory from the Set Properties and Locations window. See Using Component Selection Mode on page 127 for more information.

You can also use the 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 UNIXFILE or PATH classes.

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To log on to the Radia Packager

1 Log in as root.

2 Change your current working directory to the location of the `publishr` executable, and type **`./publishr`**.

99 In the Radia Packager Security Information dialog box, type your User ID and Password in the appropriate fields.

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** fields.

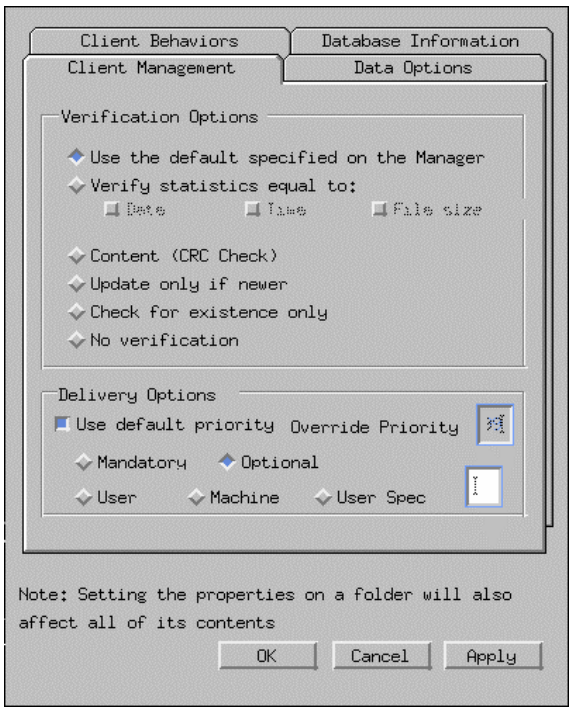
100 Click **OK**.

To access the Global Default Properties dialog box

- Go to **Edit** → **Change Global Defaults**. The Radia Packager – Global Default Properties dialog box has four tabs: **Client Management**, **Data Options**, **Database Information**, and **Client Behaviors**.

Client Management Tab

Use the **Client Management** tab to set verification and delivery options for the files or directories in the package.



Verification Options

Use **Verification Options** to specify the default actions that the Application Manager will take for the file or directories.

Your selections in this dialog box set the variable ZRSCVRFY in the base instance of the UNIXFILE class. Use the System Explorer to view, or modify, this variable.

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Table 11: Verification Options

Verification Option	Usage	ZRSCVRFY Settings
Use the default specified on the Manager	Select this option to inherit verification options for the files from the UNIXFILE class' base instance in the Radia Database.	ZRSCVRFY=Y
Verify statistics equal to	Select this option so that the Application Manager checks the selected statistics [Date (D), Time (T) or File Size (S)] for the files on the computer. The files are deployed from the Radia Database (or Staging Server) if the statistics of the file on the computer are different from the statistics for this file.	ZRSCVRFY=D ZRSCVRFY=T ZRSCVRFY=S ZRSCVRFY=Y (to check date, time and size)
Content (CRC Check)	Select this option to perform content CRC checking for the resource. This populates the ZRSCCRC attribute of the resources UNIXFILE class. Note: Use of Content CRC checking is a time consuming process and should be used sparingly.	ZRSCVRFY=Y
Update only if newer	Select this option so that the files are deployed if the files in the Radia Database (or Staging Server) have a later date/time stamp than the ones on the subscriber's computer.	ZRSCVRFY=U

Check for existence only	Select this option so that the files are deployed if they are not on the subscriber's computer. No action is taken if the files already exist on the subscriber's computer, even if the files' statistics differ from those in the Radia Database.	ZRSCVRFY=E
No verification	Select this option so that 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 variables in the base instance of the UNIXFILE class. Use the System Explorer to view, or modify, the appropriate variables.

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Table 12: Delivery Options

Delivery Option	Usage	Variable 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 subscriber's 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 subscriber's 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 subscriber's 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.		
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.	ZCONTEXT=U
Machine	Select **Machine** to indicate that the file will be deployed to all users of the computer.	ZCONTEXT=M
User Spec	*This option is for future use.*	*This option is 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.

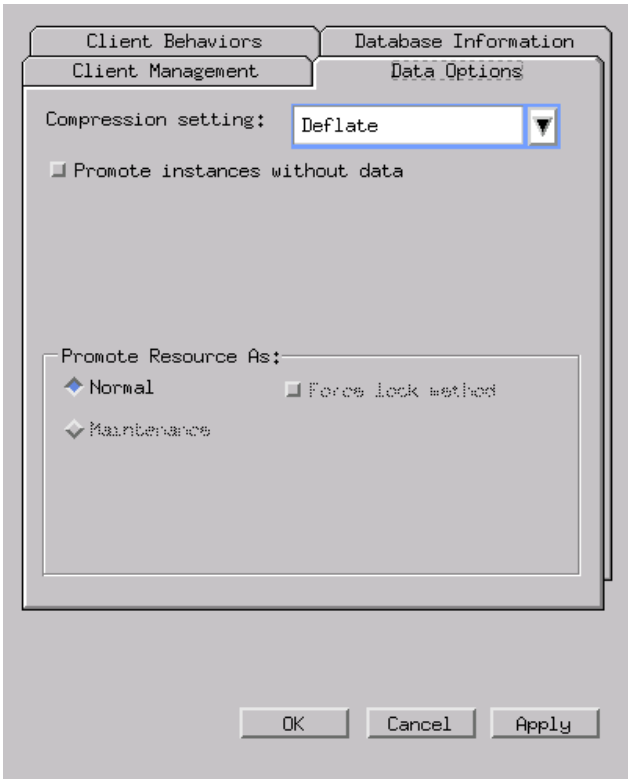


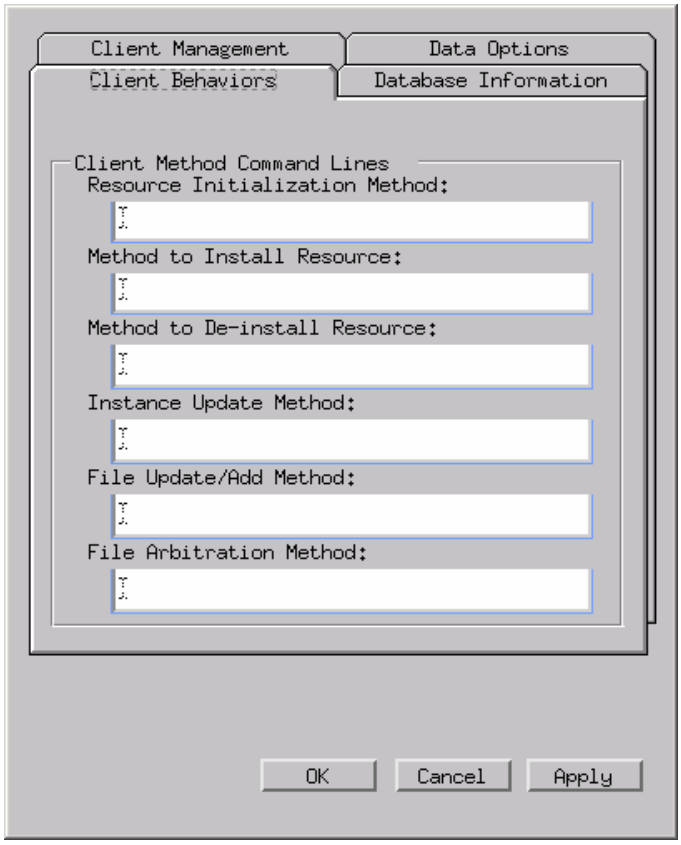
Table 13: Data Options

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

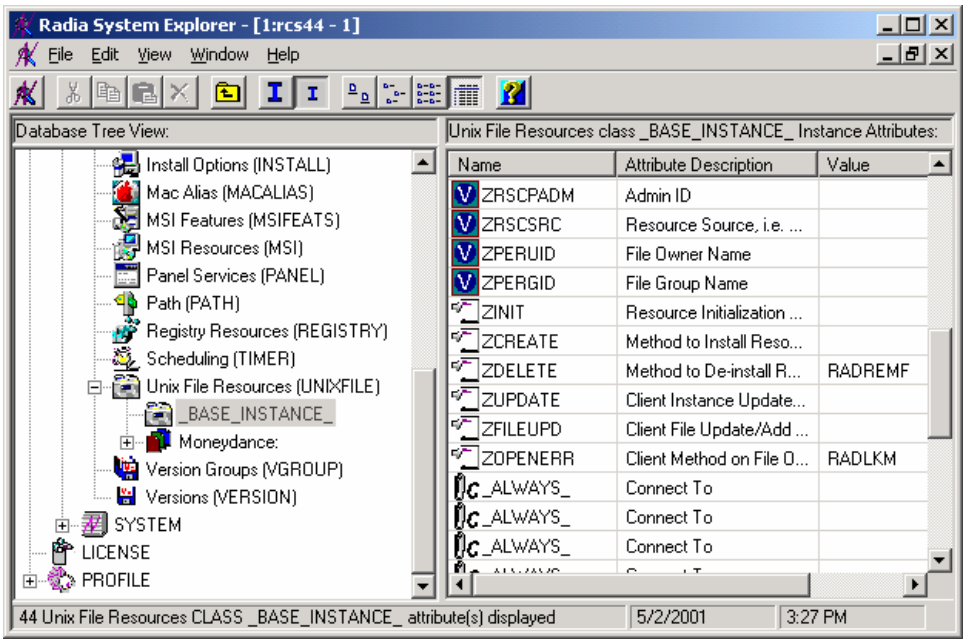
Promote instances without data	Select this check box to indicate that the files should not be transferred to the Radia Database as part of the package. Only the instances representing the files are included in the package. The data remains in compressed format in the IDMDATA location on your computer. See Chapter 2, Installing the Radia Clients for more information. You can manually place files on Staging Servers if you have connectivity to the Staging Server and do not want copies of the files in the Radia Database.
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	Select this option to indicate that the files are maintenance components for the Application Manager software.
Force lock method	Select this check box to force the use of the **locked file method** for deploying the files. If files are in use on the subscriber's computer when Radia attempts to deploy new copies of the files, the locked file method is normally used to deploy the files. If necessary, the files are decompressed and stored locally in a directory. The Client Connect process forces a restart when it ends and the files are deployed to their correct locations during startup.

Client Behaviors Tab

Use the **Client Behaviors** tab to specify default methods (or programs) that Radia executes on the subscriber's computer.



The command lines that you type in this dialog box are stored in variables in the UNIXFILE class instances in the SOFTWARE domain. The figure below displays a command line stored in a FILE class instance that was typed into the **Method to De-install Resource** text box.



This figure shows the System Explorer, which is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Table 14 below, describes the fields available on the Client Behaviors tab.

Table 14: Client Behaviors

Client Behaviors	Usage
Resource Initialization Method (Variable in database: ZINIT)	Type the method to run when the files are stored on the subscriber's computer.
Method to Install Resource (Variable in database: ZCREATE)	Type the method to run after the file is stored on the computer. This is used if some processing is required to enable the file to be used on the computer.
Method to De-install Resource (Variable in database: ZDELETE)	Normally, files are removed if the subscription to the software is cancelled. If a file, such as shared objects, is not supposed to be deleted from the subscriber's

	computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for **Method to De-install Resource**.
Instance Update Method (Variable in database: ZUPDATE)	Type the method to run when the instance is modified on the computer, after the file has been deployed.
File Update/Add Method (Variable in database: ZFILEUPD)	Type the method to run when the file is new or has been updated. The method executes just before the file is deployed to the computer.
File Arbitration Method (Variable 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 file that is going to replace it, and then determines which file to keep.

Database Information Tab

Use the **Database Information** tab to specify where the instances for the selected files or directories will be stored in the Radia Database.

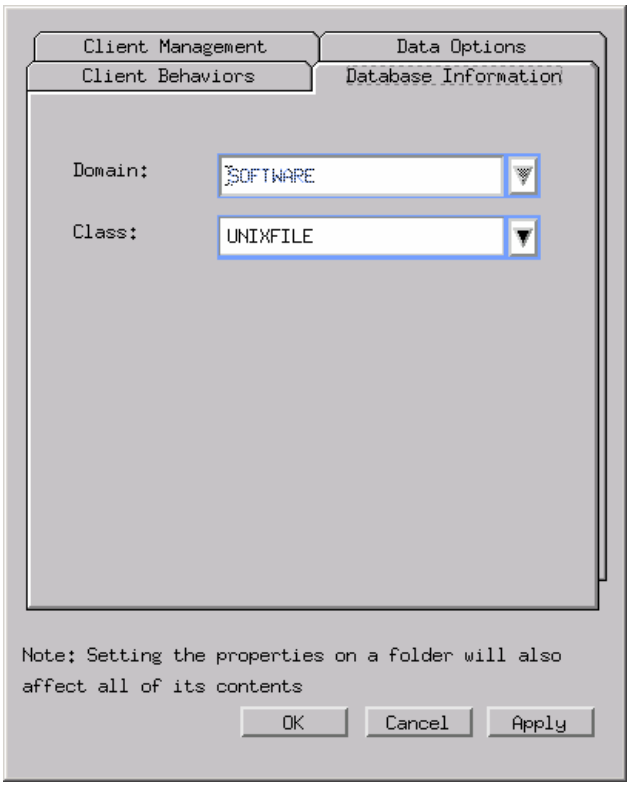


Table 15 below, describes the fields available on the Database Information tab.

Table 15: Database Information

Database Information	Usage
Domain	Domain that stores the instance. This is normally the SOFTWARE domain unless you have customized the Radia Database with proprietary domains. As shipped from HP, the default domains are ADMIN, AUDIT, NOVADIGM, PATCH, POLICY, SOFTWARE, and SYSTEM. See Chapter 1, Introduction for more information about the domains.
Class	Component class that stores the instance for this file or directory.

UNIX File Resources (UNIXFILE)

During the publication process, the UNIXFILE attributes are defined. These attributes define the owner and group associations and permissions of each published resource. Each package published has a corresponding UNIXFILE instance within the PRIMARY domain. Use the System Explorer to view and modify these attributes.

The following instructions use the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To view the UNIXFILE class instances using the System Explorer

1 Go to **Start** → **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 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.

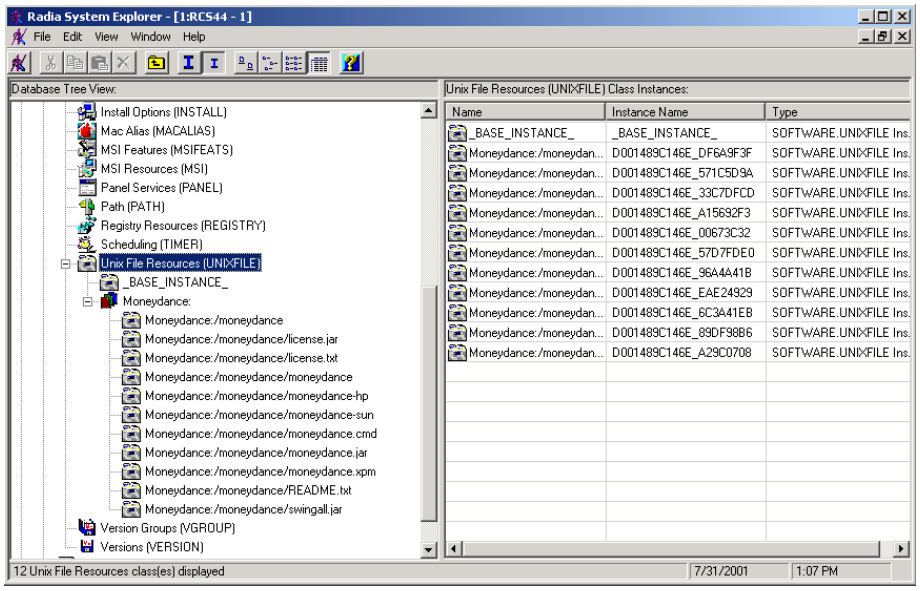
101 Double-click **PRIMARY**.

102 Double-click **SOFTWARE**.

103 Double-click **Unix File Resources (UNIXFILE)**.

104 Double-click the appropriate application. The attributes for the UNIXFILE instances for that application appear in the list view.

To change any instance attribute, double-click the attribute name in the list view. Make your desired changes in the box that opens, and click **OK** when finished.



Published Owner, Group, and Permission Considerations

The UNIXFILE class contains the attributes ZPERUID and ZPERGID. They define the user ID and group association of the promoted resource. These attributes are populated during the publishing session and reflect the user ID and group association of the resources being promoted. In addition, permission characteristics are captured during publishing and stored in the UNIXFILE.ZRSCRASH attribute. These attributes can be changed using the System Explorer.

Table 16: Attributes Exclusive to the UNIXFILE Class

Attribute	Description
ZPERUID	UNIX user ID associated with the promoted resource. The resource will be owned by this user ID when deployed, providing the Application Manager is run by root and the user ID exists on the client workstation.
ZPERGID	UNIX group ID associated with the promoted resource. The resource will be associated with this group when deployed, providing the Application Manager is run by root and the group exists on the client workstation.

Attribute	Description
ZRSCRASH	This should be a four-digit octal notation of the managed resources permissions (example: 7555). This is populated during the publishing session based on the characteristics of the published resources.

If the Application Manager is run as a non-root user ID:

- All deployed resources will be associated with the user ID and group of the user ID who is running the Application Manager.
- During publishing, the owner and group of the resource is stored in the UNIXFILE instance data. The owner and group attributes within the instance are only applied if the Radia client is run as root for only root has the ability to perform changes in owner and group characteristics.
- Radia client capabilities are limited to the permission constraints of the current user ID and group membership for the Unix user ID running the connect.
- Radia will be unable to deploy to directories where the directory permissions prohibit the non-root user and or group membership to write.
- Radia may be unable to set permissions on resources placed under Radia management that are already on the client workstation though owned by a different UID and/or GID.
- Radia will be unable to launch client methods requiring root authority.

If the Application Manager is run as root and:

- If the owner name of the resource, as defined in ZPERUID, and the user ID exist on the client workstation, the resource will be owned by the UNIX user ID specified.
- If the group name of the resource, as defined in ZPERGID, and the group exist on the client workstation, the resource will be associated with the UNIX group specified.

To prevent security breaches please note the following:

If the owner of a resource, as defined in ZPERUID, does not exist on the client workstation, the owner designation of the managed resource will be set to "nobody" (uid 60001).

If the group of a resource, as defined in ZPERGID, does not exist on the client workstation, the group designation of the managed resource will be set to "nobody" (gid 60001).

The Radia Packager Toolbar

The Radia Packager toolbar, as seen in Figure 4 below, is used to navigate among the various completed publishing sections during Installation Monitor Mode, which is available for the Windows version of the Radia Packager. The buttons available for use with the UNIX version of the Radia Packager platform will be highlighted.

Figure 4: Radia Packager toolbar.

Using Component Selection Mode

In Component Selection Mode, you select the individual components that make up the application, such as files, directories, and links to create a package.

Publishing in Component Selection Mode involves three phases:

1 **Defining** the application's operating system requirements.

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

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

Prerequisites

Before publishing your application in Component Selection Mode:

- Install the target application on your packaging machine. This ensures that the files you need to select reside on the computer.

Publishing

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

In this example, we publish the shareware application, Moneydance. You can substitute another application or file in its place.

Use this example to become familiar with Component Selection Mode. However, please remember that there are many variables when publishing applications.

For the following example to work as shown, be sure to download and install Moneydance on your administrator computer. Moneydance is available at www.moneydance.com.

Task 1 Logging On to Radia Packager

1 Change your current working directory to the location of the `publishr` executable, and type **`./publishr`**.

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

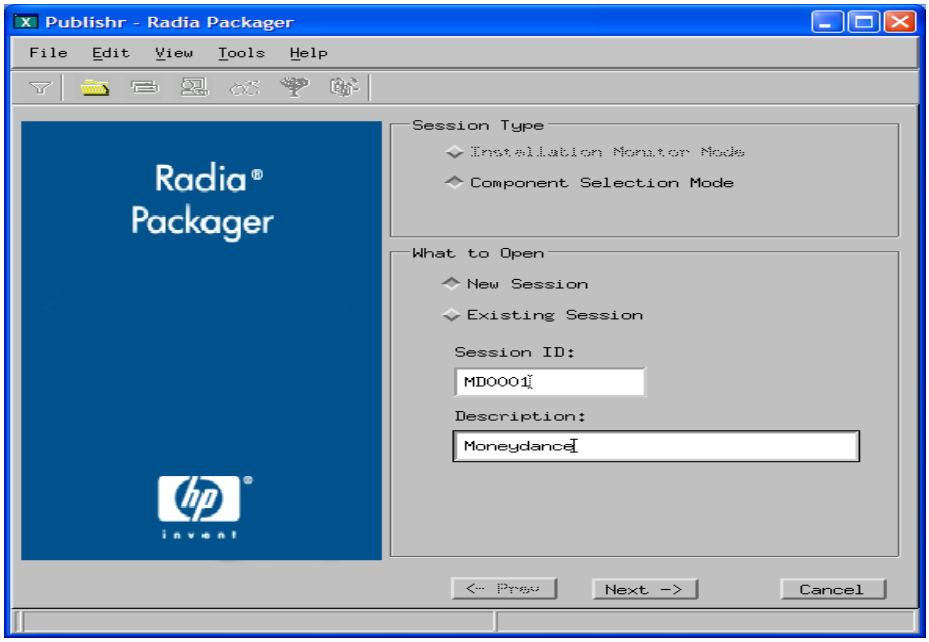
The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This may have been changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the **New Password** and **Verify New Password** fields.

106 Click **OK**.

The Open Publishing Session window opens.

Task 2 Completing the Open Publishing Session Window

Use the Open Publishing Session window to select the publishing mode and enter information to identify the publishing session.



At the end of a **publishing session**, you will have a Radia package – a unit of distributable software and/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 section describes the options on the Open Publishing Session window.

We recommend that you review the Global Defaults prior to beginning any Publishing Session. See Setting Default Properties beginning on page 112.

Session Type Area

- **Installation Monitor Mode**
  Not available at this time.
- **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 subscriber's computer. In this mode, you identify the files that are packaged and then promoted to the Radia Database.

What to Open Area

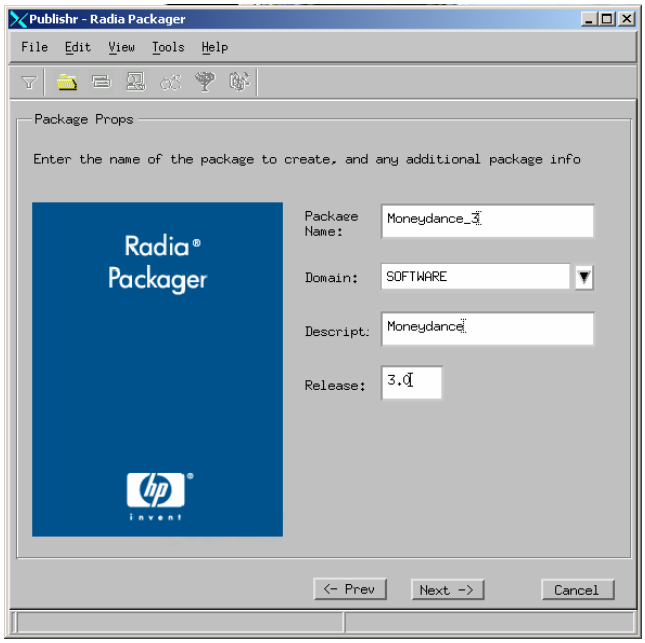
- **New Session**
  Select **New Session** to begin a new publishing session. You must complete the **Session ID** and **Description** fields.
- **Existing Session**
  Select **Existing Session** to resume the previous session.
- **Session ID**
  Type a *unique* identifier that is one to six characters long.
- **Description**
  Type a description of the session.

In the Moneydance example, we are using **Component Selection Mode** to begin a **New Session**. The **Session ID** is **MD0001** and we've described the session as Moneydance.

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

Task 2 Entering Package Properties

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



- **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.

  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 for more information.

- **Domain**
  Select the domain to store the instance in. This is normally the SOFTWARE domain unless you customized the Radia Database with proprietary domains. As shipped from HP, the default domains are ADMIN, AUDIT, NOVADIGM, PATCH, POLICY, SOFTWARE, and SYSTEM.

See Chapter 1, Introduction for more information about the domains.

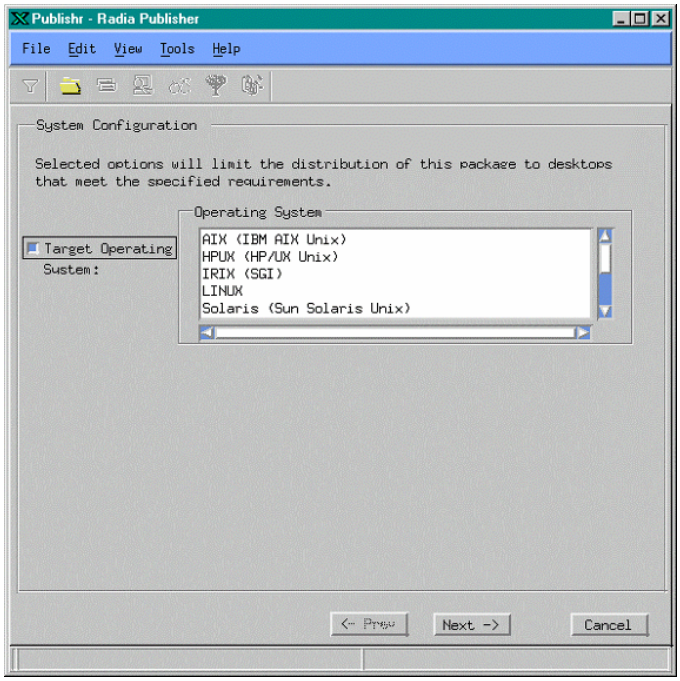
- **Description**
  Type a description for the package.
- **Release**
  Type the release number of the software.

In the Moneydance example, we named the package **Moneydance_3** and gave it a description and release number: Moneydance, and 3.0, respectively.

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

Task 3 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. If none of the options is selected, the package will be available to all eligible subscribers.



- **Target Operating System**
  Select this check box to make the Operating System options available. Then, select the operating systems for which this package applies.
  - 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.

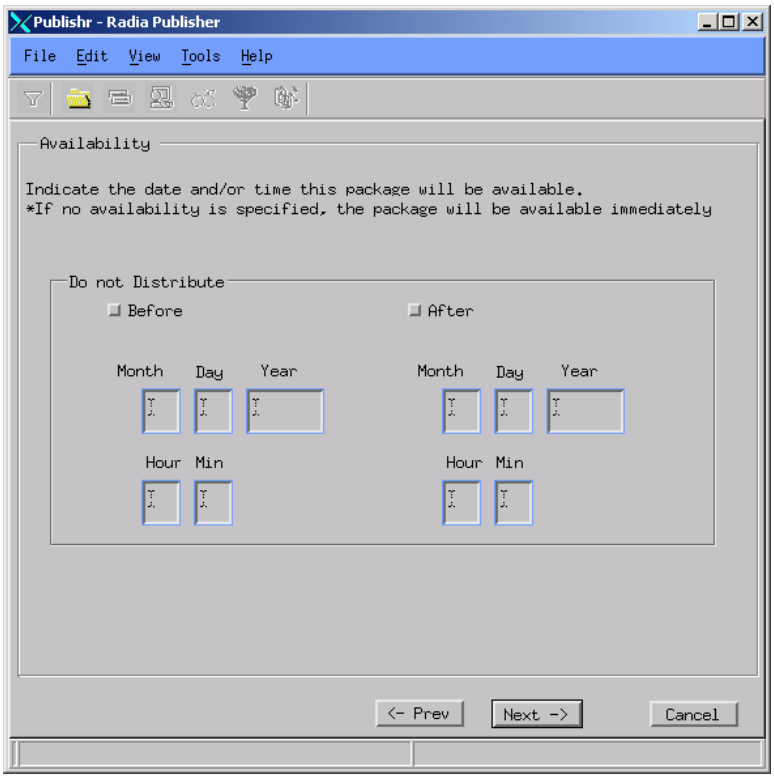
In the Moneydance example, we set the target operating system to Solaris.

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

Task 4 Setting Date and Time Constraints

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

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.



- **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** fields 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** fields to specify the date and time.

In the Moneydance example, we accepted the default settings so that the package will be available for distribution as soon as we promote it to the Radia Database.

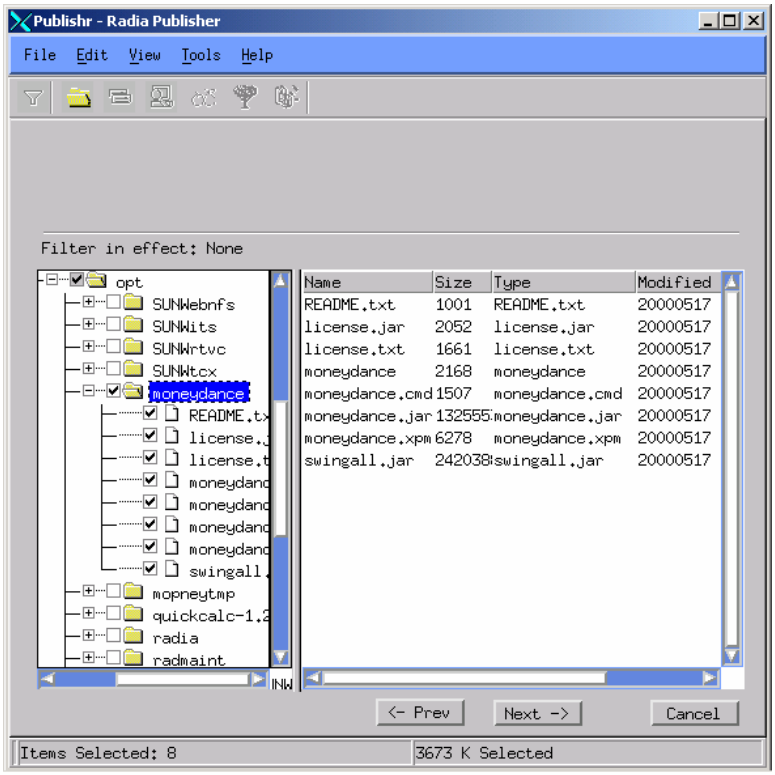
Click **Next** to go to the Select Files to Be Published window.

Task 5 Selecting the Files to Publish

Use the Select Files to be Published window to select all files that need to be included in the package.

To select the files to publish

- Navigate through your file system and select the files or directories to be included in the package. Click a check box again to clear a selection.



In the Moneydance example, we selected the directory `/opt/moneydance` that contains the program files.

The file selection window displays the files available in order by:

- An alphabetized listing of directories.
- Then, an alphabetized listing of files.
- An alphabetized listing of UNIX links.

Re-size the file selection window by positioning your mouse over the vertical bar separating the two windows, clicking and dragging to the left or right.

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

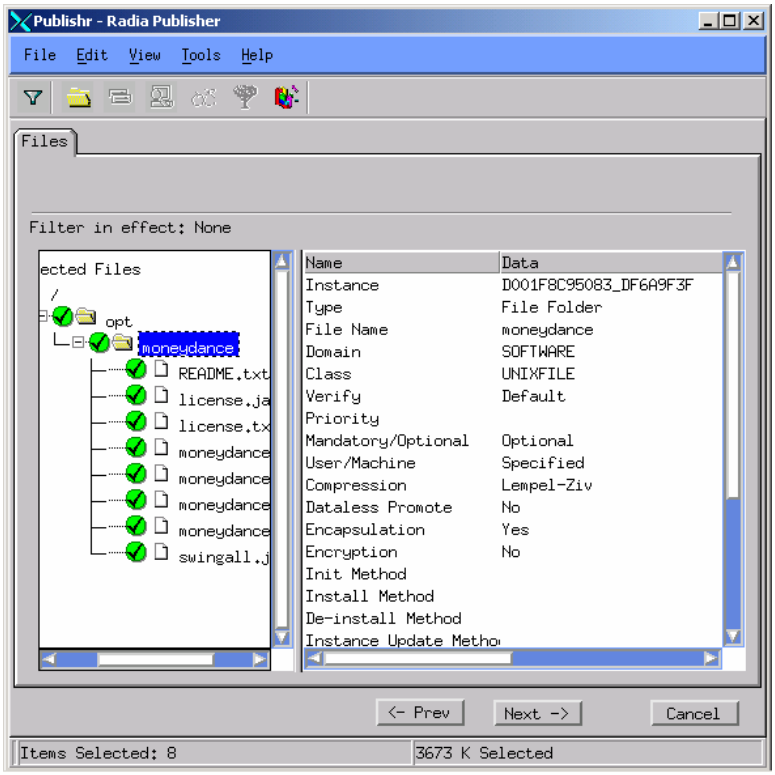
Task 6 Viewing File Properties and Locations

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

To view the selected files and directories

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 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 112. In Setting Properties and Locations on page 136, you will learn how to modify these settings, if necessary.



To filter the displayed files and directories

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. If the filter string is found anywhere in the file or directory name, it is considered a match.

107 Click **OK**.

To remove a filter

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

2 Click **Clear**.

108 Click **OK**.

Click **Next** to continue.

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

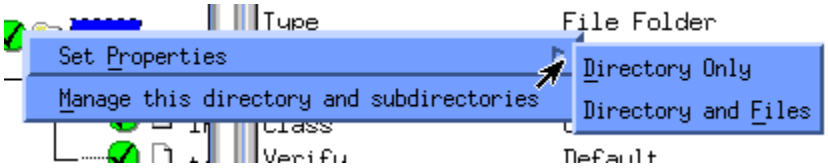
Task 7 Setting Properties and Locations

Use the Instance Properties dialog box to modify the properties of the files and directories in the package. A file's properties 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 Properties on page 112.

To access the Instance Properties dialog box

1. Right-click the file or directory whose attributes you want to set.
2. From the shortcut menu, select **Set Properties**. If you selected a directory, 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, Database Information, and Client Behaviors.

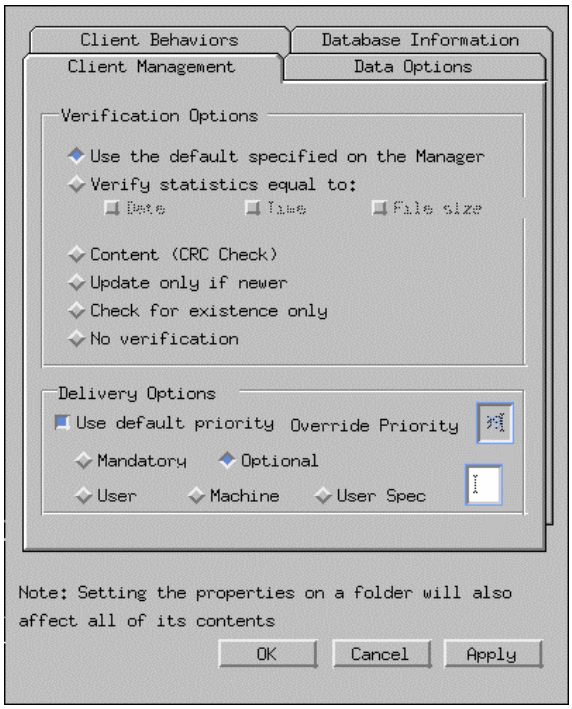


Client Management Tab

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

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

- Use the **Verification Options** to specify the actions that the Application Manager will take for this file or directory.
- Use the **Delivery Options** to specify delivery options, such as the order in which files are deployed.



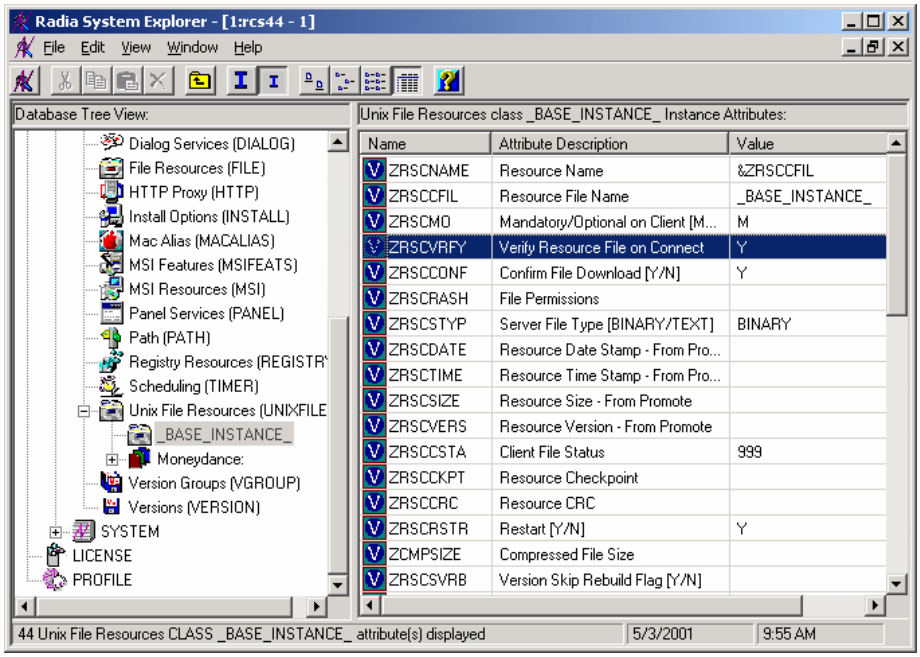
Verification Options

- **Use the default specified on the Manager** (default)

  Select this option so that verification options for these files or directories are inherited from the base instance of the UNIXFILE class.

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

The following figure and bullet points refer to the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.



- **Verify statistics equal to**
  Select this option so that the Application Manager checks the selected statistics (**Date**, **Time**, or **File Size**) for the files or directories on the computer. The files or directories are deployed from the Radia Database or Staging Server if the statistics of the files or directories on the computer are different from the statistics for these files or directories. You can also use System Explorer to set this option: ZRSCVRFY=D, ZRSCVRFY=S, ZRSCVRFY=T, or ZRSCVRFY=Y.

- **Content (CRC Check)**
  Select this option to perform content CRC checking for the resource. This populates the ZRSCCRC attribute of the resource's UNIXFILE class. ZRSCVRFY is set to Y.

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

- **Update only if newer**
  Select this option so that these files or directories are deployed if the files or directories in the Radia Database (or Staging Server) have a later date/time stamp than those on the subscriber's computer. You can also use the System Explorer to set this option: ZRSCVRFY=U.

- **Check for existence only**
  Select this option so that these files or directories are deployed if they are not on the subscriber's computer. No action is taken if the files or directories already exist on the subscriber's computer, even if the statistics differ from those in the Radia Database. You can also use the System Explorer to set this option: ZRSCVRFY=E.
- **No verification**
  Select this option so that the files are deployed the first time the application is deployed. No subsequent action is taken. You can also use the 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 the 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 the System Explorer to set this option: ZRSCPRI=1.

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

- **Mandatory**
  Select this option to indicate that these files or directories are critical to the application. If there is not enough space on the subscriber's computer for the entire application, Radia will deploy *only* mandatory files. If there is not enough space for the mandatory files, then the application is not deployed at all. You can also use the System Explorer to set this option: ZRSCMO=M.
- **Optional** (default)
  Select this option to indicate that files or directories are not critical to the application. If there is not enough space on the subscriber's computer for the entire application, Radia will *not* deploy optional files. You can also use the 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 or 2000.

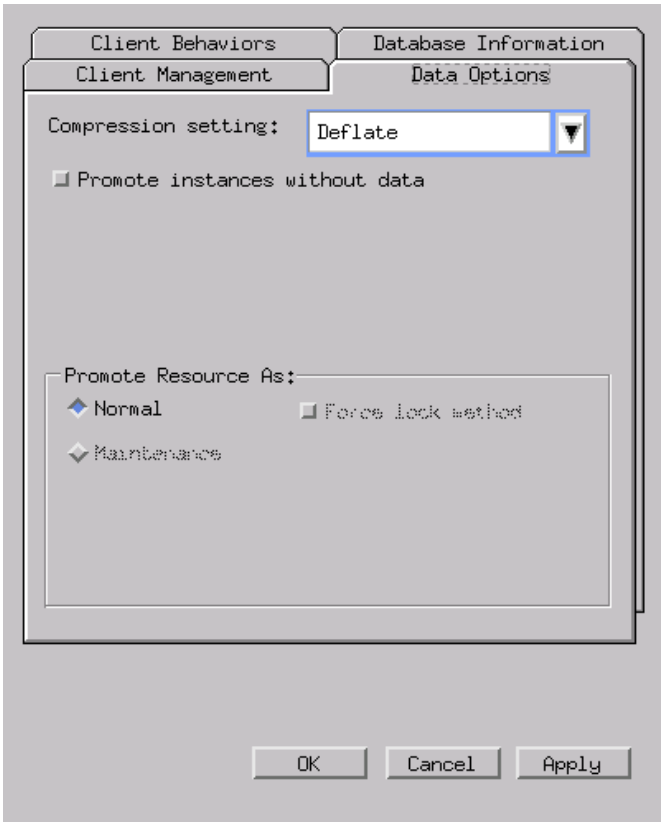
- **User**
  Select **User** if you want to indicate that the file will be deployed only to the subscriber logged on when the application is initially deployed. You can also use the System Explorer to set this option: ZCONTEXT=U.

- **Machine**
  Select **Machine** to indicate that the file will be deployed to all users of the computer. You can also use the System Explorer to set this option: ZCONTEXT=M.
- **User Spec**
  This option is reserved for future use.

In the Moneydance 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 directories that you will be distributing.



- **Compression setting**
  If necessary, select the compression setting for storing these files or

directories in the Radia Database. Compression minimizes the time required to transmit it and the amount of disk space required to store it.

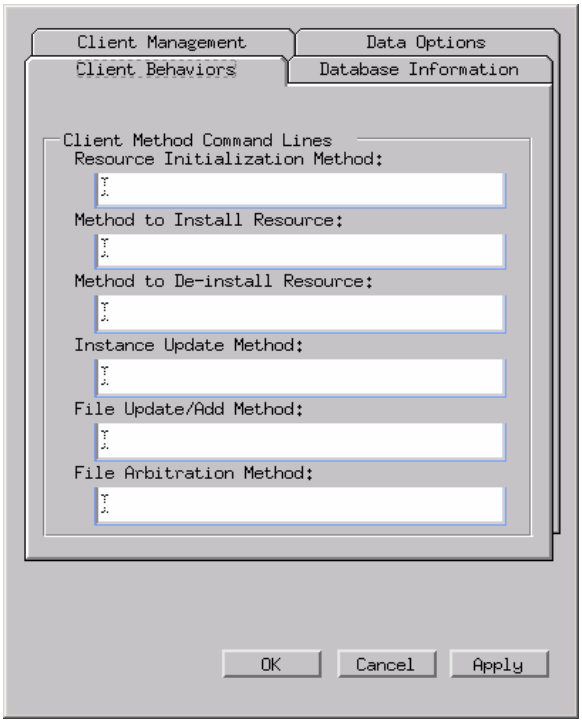
- Select **Deflate** for the most efficient compression, which produces smaller compressed images. This is the default setting.
- Select **None** if the files or directories are already compressed. If you are packaging an application that contains one or more compressed files, do *not* have the Radia 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 these files or directories should not be transferred to the Radia Database as part of the package. Only the instance representing the file is included in the package. The data remain in compressed form in the IDMDATA location on your computer. See Chapter 2, Installing the Radia Clients for more information. You can manually place files on a Staging Server if you have connectivity to it and do not want a copy of the file in the Radia Database.

Promote Resource As

- **Normal**
  Select this option to indicate that these files or directories are to be deployed as part of an application.
- **Maintenance**
  Select this option to indicate that these files or directories are a maintenance component for the Application Manager software.
- **Force lock method**
  Select this check box to force the use of the **locked file method** for deploying these files or directories. If the files or directories are in use on the client computer when Radia attempts to deploy new copies of the files or directories, the locked file method is normally used to deploy the files or directories. If necessary, these files or directories are decompressed and stored locally in a directory. The Client Connect process forces a restart when it ends and the files or directories are deployed to their correct location during the startup.

Client Behaviors Tab

Use the **Client Behaviors** tab to specify methods (or programs) that Radia executes on the subscriber's computer.



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

Client Method Command Lines

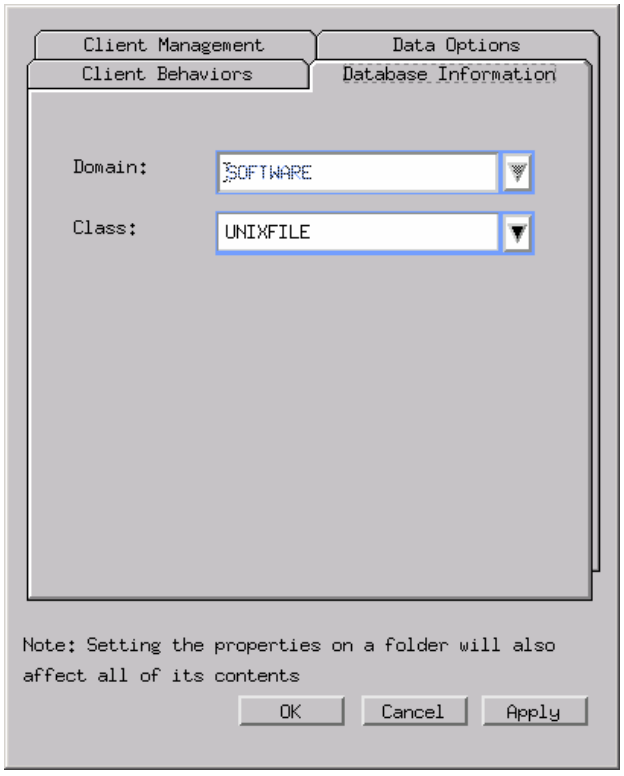
- **Resource Initialization Method** (Variable in Database: ZINIT)
  Type the method to run when the files or directories are stored on the subscriber's computer.
- **Method to Install Resource** (Variable in Database: ZCREATE)
  Type the method to run after the file is stored on the computer. This is used if some processing is required to enable the file to be used on the computer.
- **Method to De-install Resource** (Variable in Database: ZDELETE)
  Normally, files are removed if the subscription to the software is cancelled. If a file, such as a shared object file, should not be deleted from the subscriber's computer, even if the subscription to the software is cancelled, type **_NONE_** (with the underscores) as the value for **Method to De-install Resource**.

- **Instance Update Method** (Variable in Database: ZUPDATE)
  Type the method to run when the instance is modified on the computer, after the file has been deployed.
- **File Update/Add Method** (Variable in Database: ZFILEUPD)
  Type the method to run when the file is new or has been updated. The method executes just before the file is deployed to the computer.
- **File Arbitration Method** (Variable in Database: ARBITRAT)
  Type the method to run if files or directories are about to be replaced. This method examines the version information of the files or directories that exist and the files or directories that are going to replace it, and then determines which to keep.

In the Moneydance example, we did not specify any methods.

Database Information Tab

Use the **Database Information** tab to specify where the instance for the selected files or directories will be stored in the Radia Database.



- **Domain**
  Select the domain in which to store the instance. This is normally the SOFTWARE domain unless you added proprietary domains to the Radia Database. The default domains are ADMIN, AUDIT, NOVADIGM, PATCH, POLICY, SOFTWARE, and SYSTEM. See Chapter 1, Introduction for more information about the domains.
- **Class**
  Select the component class to store the instances for these files or directories in.

In the Moneydance example, we accepted the default settings for the domain and class.

Click **OK** to return to the Set Properties and Locations window.

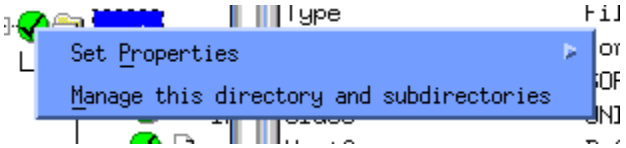
Task 8 Directory Management

From the shortcut menu select **Manage this directory and subdirectories** to control the level of directory management desired.

To establish management of specific directories

1. Expand the directory tree until the directory from where you would like to begin management is shown.
2. Right-click this directory and select **Manage this directory and subdirectories**.

   The directory and files will become highlighted, indicating Radia will now manage each directory and file beneath this directory.

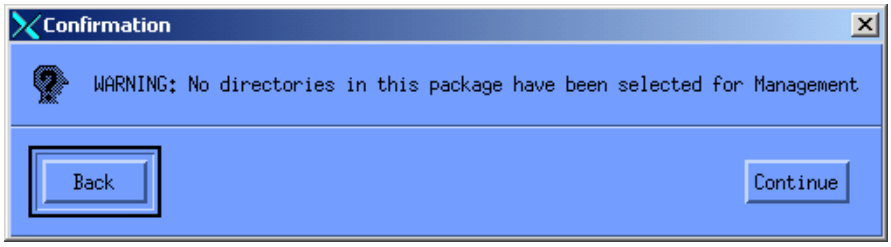


If you selected a directory you would not like managed, right-click the managed directory and select UnManage this directory and subdirectories from the shortcut menu.

The default behavior is UnManage directories and subdirectories.

You will not be able to turn this selection on for the root directory (the / directory).

If you decide to leave this option turned off, you will be prompted to confirm your decision not to explicitly manage any directories within the current package.



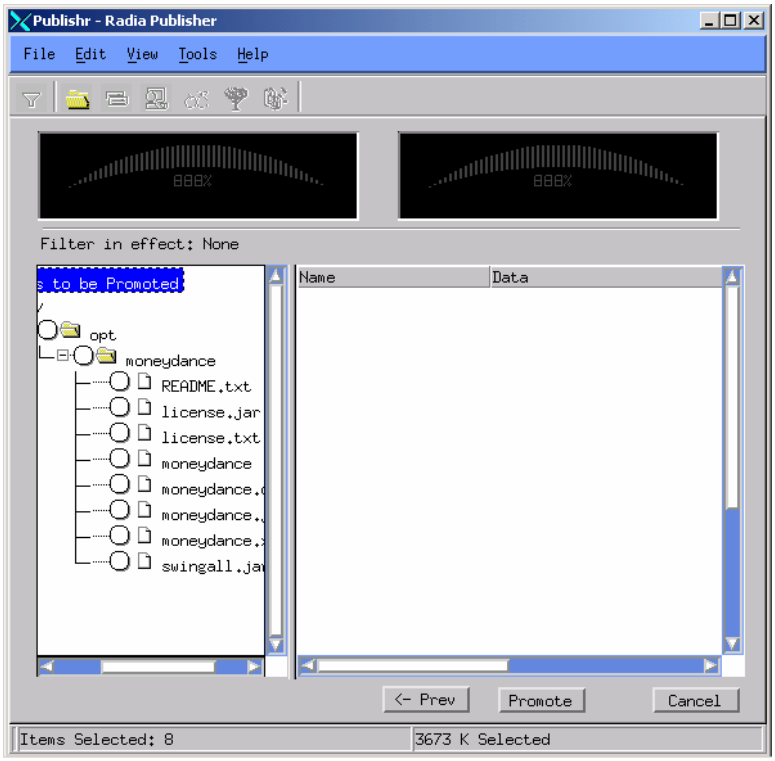
At this dialog box, you may click **Back** to choose directories for explicit management, or click **Continue** to proceed with the publishing of the package.

Task 9 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**.



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.

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

110 Click **Yes** to confirm that you want to close the Radia Packager.

In this example, Moneydance was promoted to the Radia Database. Next, use the System Explorer to create a service. For more information, see Creating a Service on page on page 148.

The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Publishing Adapter

The Publishing Adapter is a command-line alternative to using Component Selection Mode, which offers an automated, repeatable command-line process to create Radia packages and store them in the Radia Database for distribution.

The 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.

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

Radia Native Packaging

Radia Native Packaging, is a feature of the Publishing Adapter specifically designed for UNIX environments. Radia Native Packaging is a command-line driven content-publishing tool supporting native HP-UX and Solaris software. Radia Native Packaging is installed during the regular installation of the Publishing Adapter on a UNIX system.

Radia Native Packaging explores UNIX native software depots, searches for available native packages and publishes wrapped native packages to the Configuration Server. Radia Native Packaging will publish all necessary information that will allow you immediate installation of native software to end clients. When the Application Manager client is installed, a Tcl script is included in the IDMSYS directory that is required when packages published using Radia Native Packaging are deployed. For more information, refer to the *Publishing Adapter Guide*.

The 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 System Explorer to create a service. A *service* is the fundamental unit of content managed by Radia. Use the System Explorer to see services listed in the Application (ZSERVICE) class.

In this example, you will create a service using the Moneydance package that you created in Component Selection Mode, beginning on page 127.

Using the New Application Wizard to Create a Service

The following instructions use the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

Task 1 Accessing the System Explorer

1 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

2 In the System Explorer Security Information dialog box, type your User ID and Password in the appropriate fields.

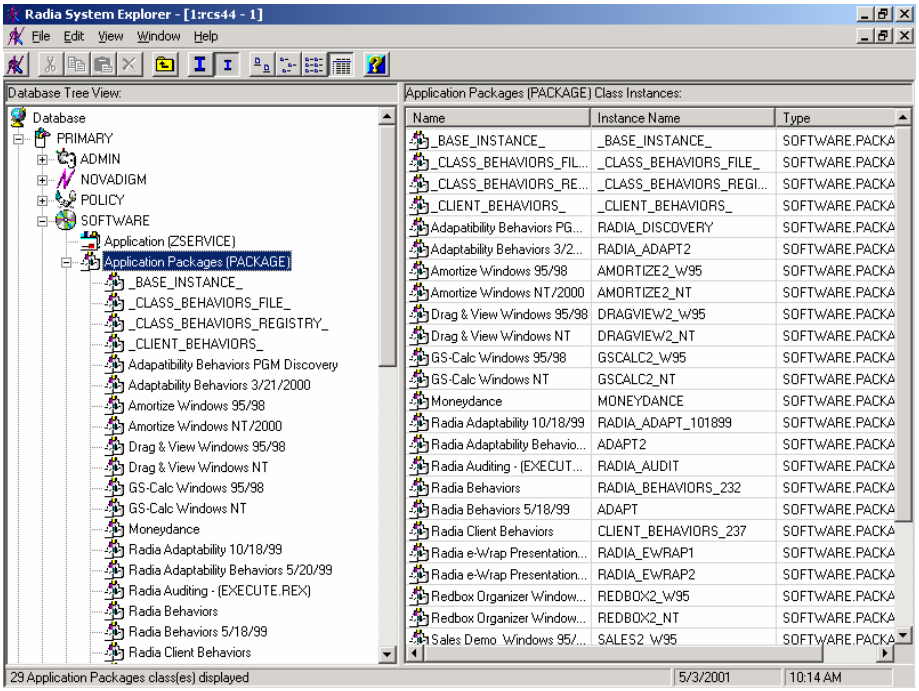
111 Click **OK**.

Task 2 Navigating 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.

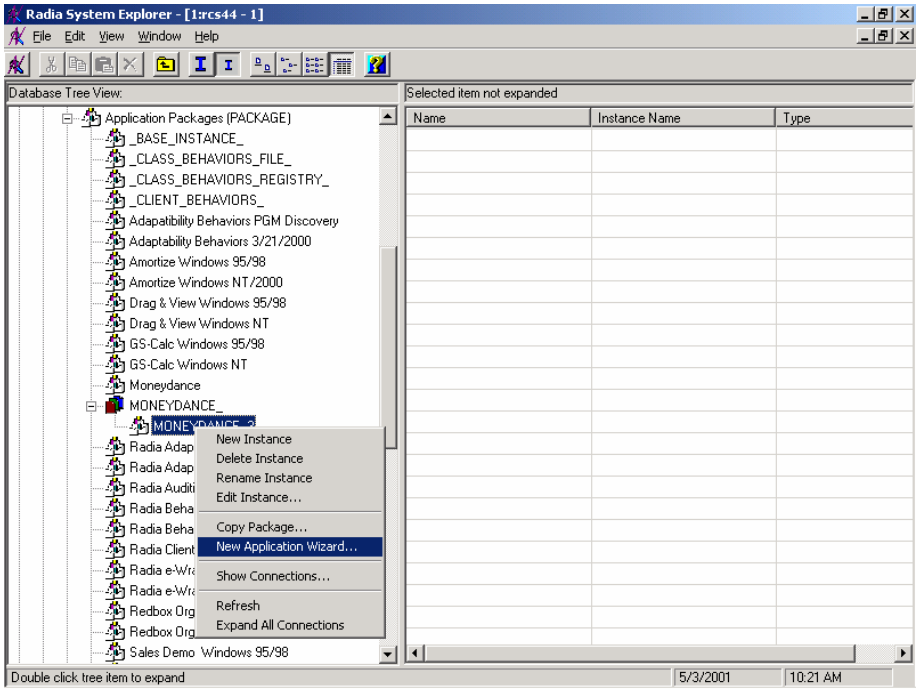
2 Double-click **SOFTWARE**. The classes of the SOFTWARE domain appear beneath its icon in the tree view and in the list view.

112 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.

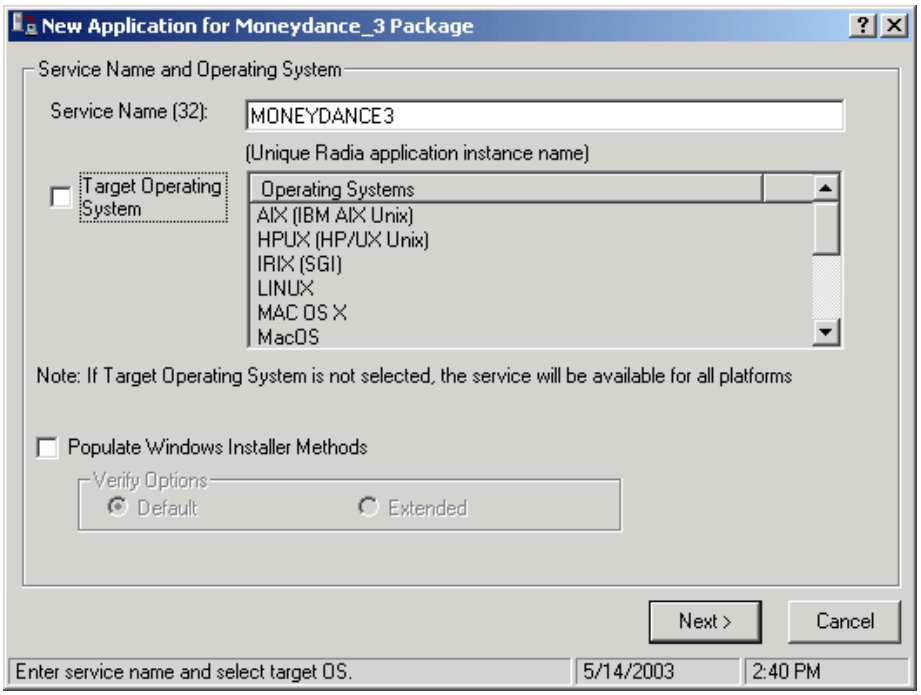


Task 10 Using the New Application Wizard to Create a Service

1 In the PACKAGE class of the SOFTWARE domain, right-click the **Moneydance** instance. A shortcut menu opens.



2 Click **New Application Wizard**.

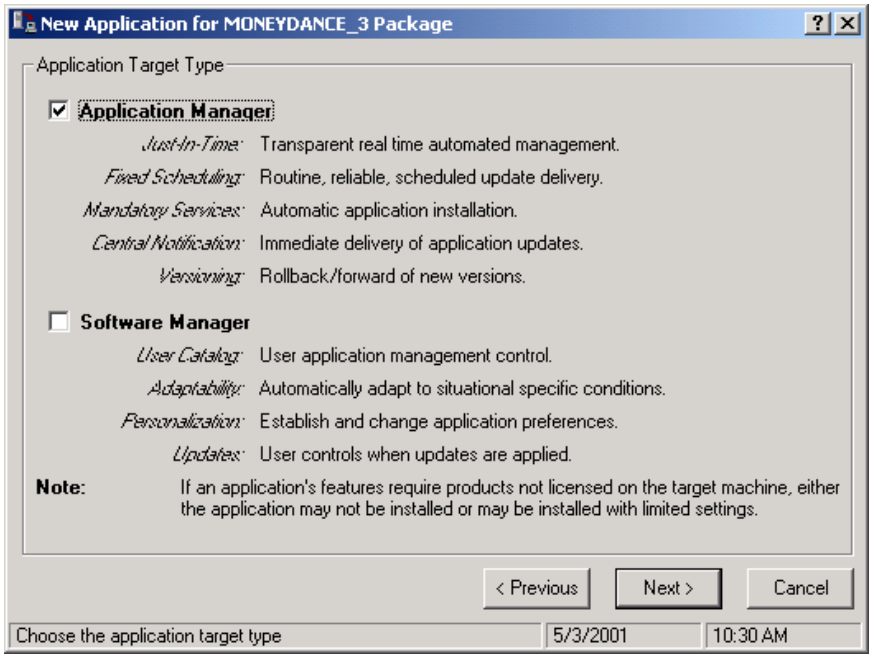


113 In the **Service Name (32)** text box, type a name, such as **MONEYDANCE3**, for the Application (ZSERVICE) instance.

114 Select the **Target Operating System** check box only if your intended target operating system appears in the list, and the specific operating system for which the package applies is selected.

115 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. This option is not applicable to UNIX-specific packages.

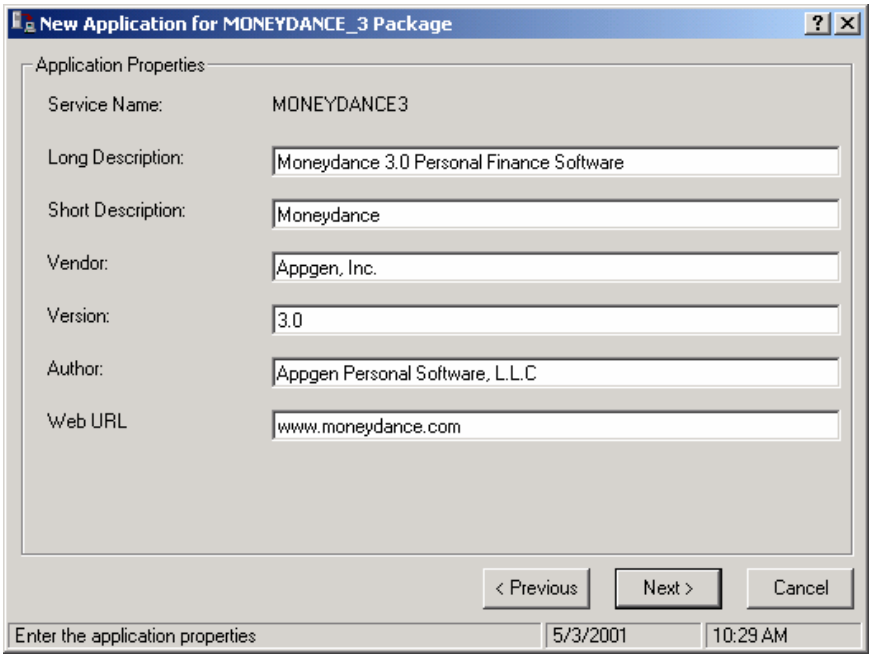
116 Click **Next** to select the application target type.



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

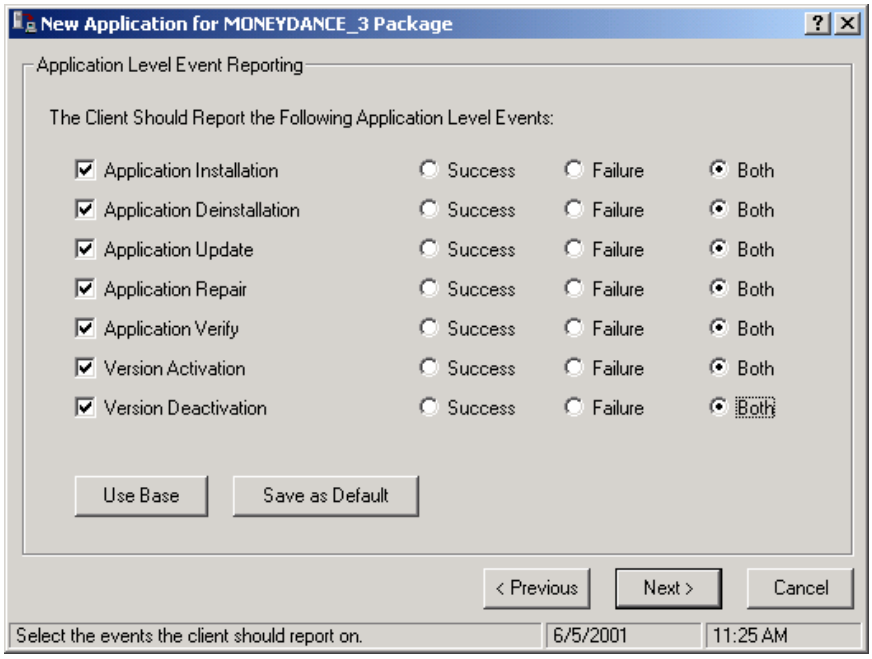
When using the Application Manager, applications *must* be mandatory in order to deploy them to your subscribers.

118 Click **Next** to enter the application properties.



119 Type the appropriate information into each Application Properties field.

120 Click **Next** to select the events that the Application Manager will report on.

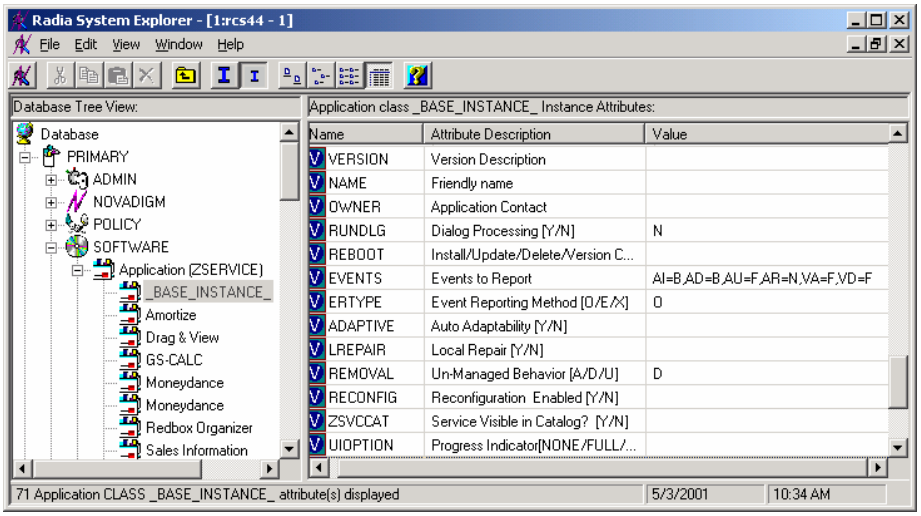


121 Click the check box for each event that you want to report on. Then, select the appropriate option button to indicate whether to report on the event's success, failure, or both.

Or

Click **Use Base** if you want to inherit the values for the ERTYPE and EVENTS variables from the base instance of the Application (ZSERVICE) instance. These variables control event reporting.

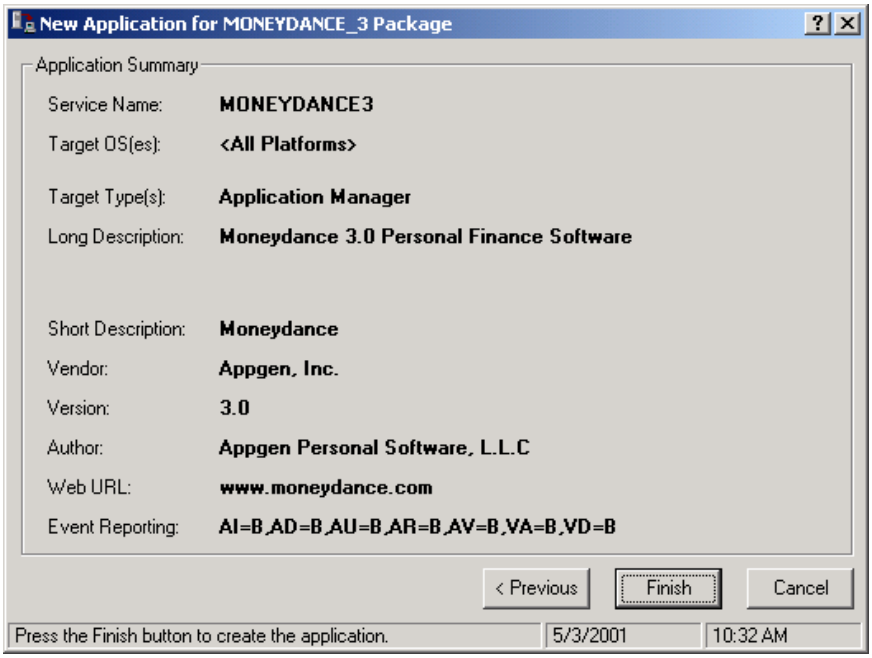
For this example, we selected every Application Event to be reported in the event of a success or failure.



For more information about these variables and the APPEVENT object, see Chapter 10, Radia Client Objects and Directories.

If you want to save the current settings as the default settings for the Application Event Panel, click **Save as Default**.

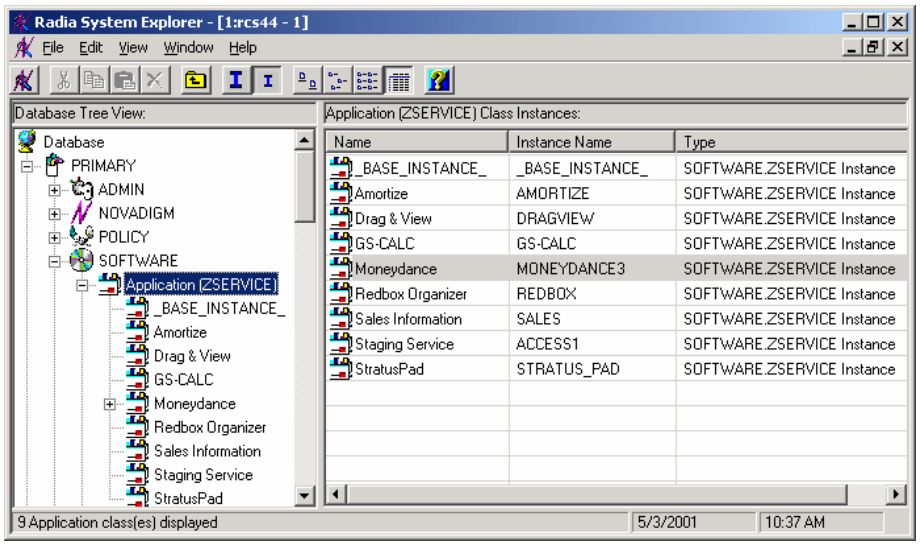
122 Click **Next** to review your selections.



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

124 Click **OK** when you are prompted with a message indicating that the application has been added. The instance appears in the ZSERVICE class.

If you want to modify any of the information that you entered in the New Application Wizard, locate the corresponding variable and change its value.



Now, you are ready to set up policies identifying *which* subscriber receives *what* software. See Chapter 6, Implementing Entitlement Policy 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 products where:

- A product may utilize more than one service-package.
- A large product may need to be split into smaller sub-services to install only specific parts of the product suite.

For detailed information on creating Radia Service Groups, refer to the *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 the component class contains the proper signature attributes as specified in the Configuration Server specifications.

Initially, to allow for the functionality of byte level differencing, the following limitations are set:

- Patches will be managed at the SOFTWARE.ZSERVICE level between PACKAGES instances that are hierarchically connected together.
- Patches can only be created between components with the same fully qualified names.
- Patches can only be created for components that contain a signature. Initially, only MD5 is supported.

Components being used for patching must be published from the same location, or computer, to qualify for byte-level differencing patching. This will populate the eight-byte CRC found in the suffix of the instance names.

For detailed information, refer to the *System Explorer Guide*.

Summary

- Publishing is the process of identifying the components of the software or content and organizing them into packages.
- Radia publishing mode: Component Selection Mode.
- To publish packages, install the Radia Packager onto a clean computer. To configure applications you must use the System Explorer.
- Install the Radia Packager onto a machine you will be using for publishing applications.
- You can use Component Selection Mode for packaging simple applications by selecting the individual components that make up the software.
- Use the Publishing Adapter as an alternative to Component Selection Mode.
- After publishing applications, use the New Application Wizard in the System Explorer to create a service—the fundamental unit of content managed by Radia.
- You can use Radia to prepare and distribute maintenance to the Application Manager.

6 Implementing Entitlement Policy

At the end of this chapter, you will:

- 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 *suggested* implementation for the 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 Application Manager. This chapter covers Implementing Entitlement Policy, assigning users to groups, and connecting applications to users.

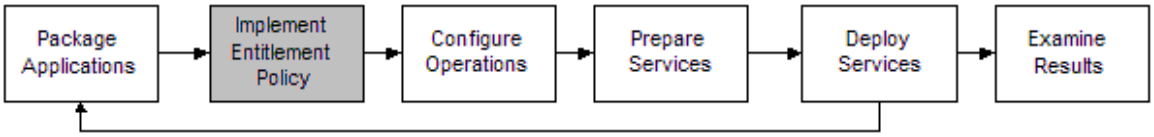


Figure 5: Tasks completed in this guide.

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:

- NT Domains
- 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 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. However, typically, you will want Radia to search your existing policy stores. This information is sent back to the Configuration Server, which determines which digital assets are to be managed for the user, group, or computer.

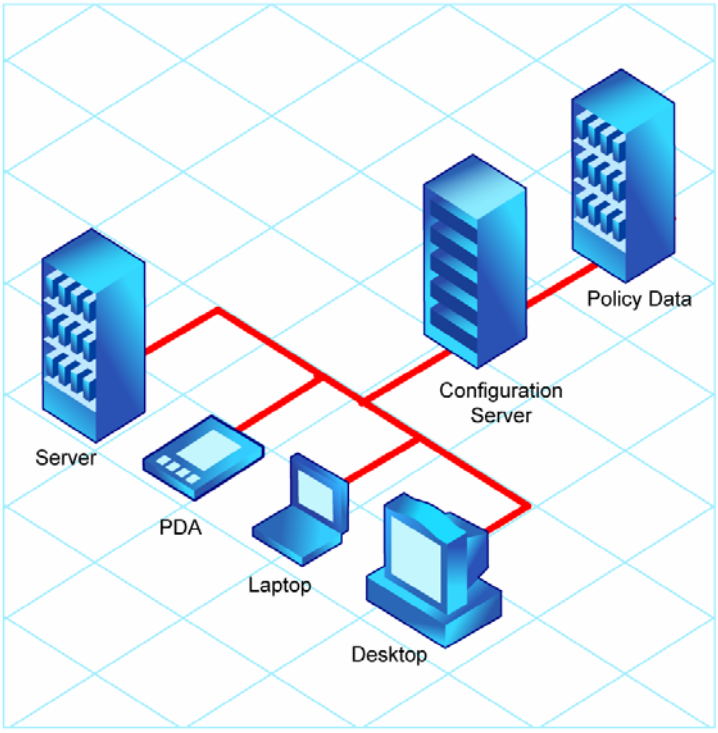


Figure 6: Retrieving policy information from an external source.

Radia also supports using multiple 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 of time. During this time, you can continue to use as many existing policy stores as necessary.

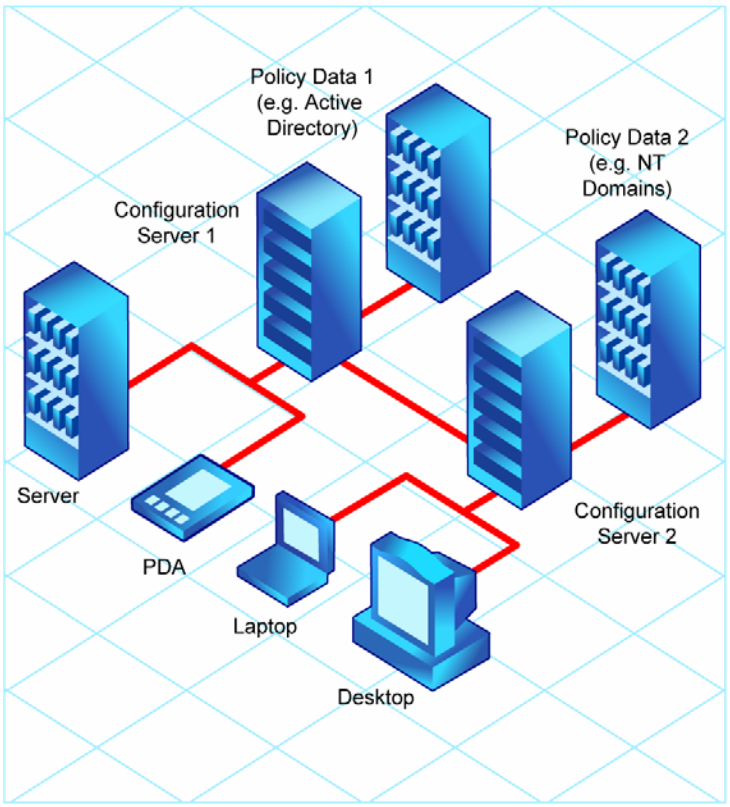


Figure 7: Retrieving policy information from multiple external sources.

Integrating with Existing External Policy

In order to use real-time policy information from an external source to manage your digital assets, Radia must communicate with your policy system. Each policy system has its own interface; some are proprietary, some are standardized.

For technical details about integrating your existing policy with Radia, see the HP OpenView web site.

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, we offer the Policy Manager. The Policy Manager is a plug-in to the Radia Integration Server (RIS) used for administration purposes such as mapping services to users in the directory tree. The Configuration Server can be configured to query the Policy Manager to determine what services should be distributed and managed for the client that is currently logged on.

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 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 their 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.

The following section uses the System Explorer, which is available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To access the POLICY domain

1 Go to **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**.

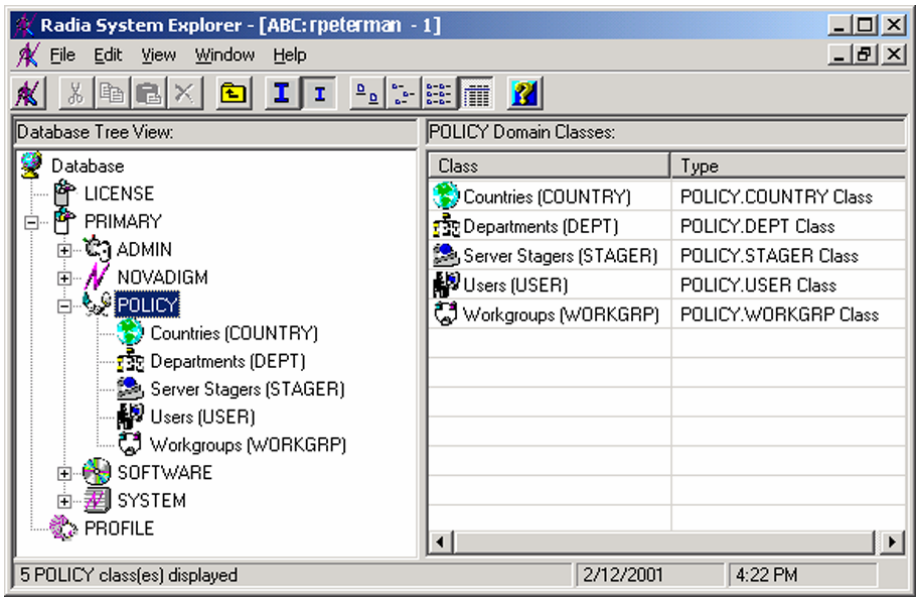
The Radia System Explorer Security Information dialog box opens.

2 If necessary, type a User ID and Password, and then click **OK**.

The 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.

125 Double-click **PRIMARY**.

126 Double-click **POLICY**.



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 17: Classes in the POLICY Domain

Class	Description	Instance Examples
Countries (COUNTRY)	Use for clock synchronizations with the Configuration Server. Do not assign services to this class.	France, Japan, Italy
Departments (DEPT)	Use to group subscribers into departments.	Finance, Customer Service, Manufacturing
Server Stagers (STAGER)	Use to define Staging Servers within your distribution network. Also, use to define storage locations on a Staging Server computer.	CDROM, Stager, Server001
Users (USER)	Use to define individual subscribers.	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. Or, if your organization is a bank, you might add classes such as BRANCHES or TELLERS to organize your subscribers.

Refer to the *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 System Explorer to create a new user (Robin) in the USER class.

The following instructions use the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To create a new user

1 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

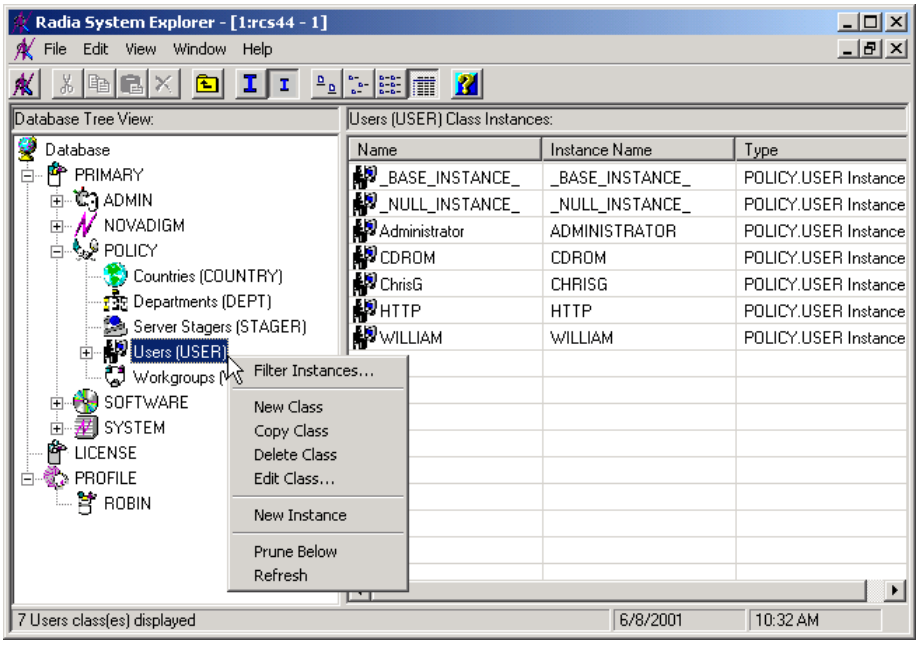
The System Explorer Security Information dialog box opens.

2 If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.

127 Double-click **PRIMARY**.

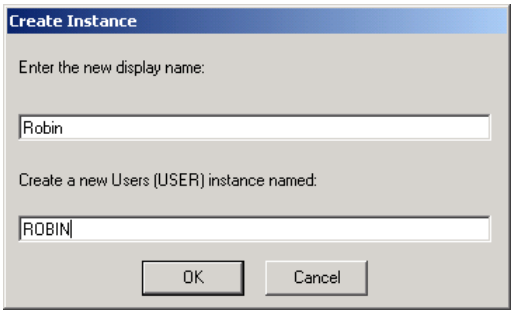
128 Double-click **POLICY**.

129 Right-click **Users (USER)**.



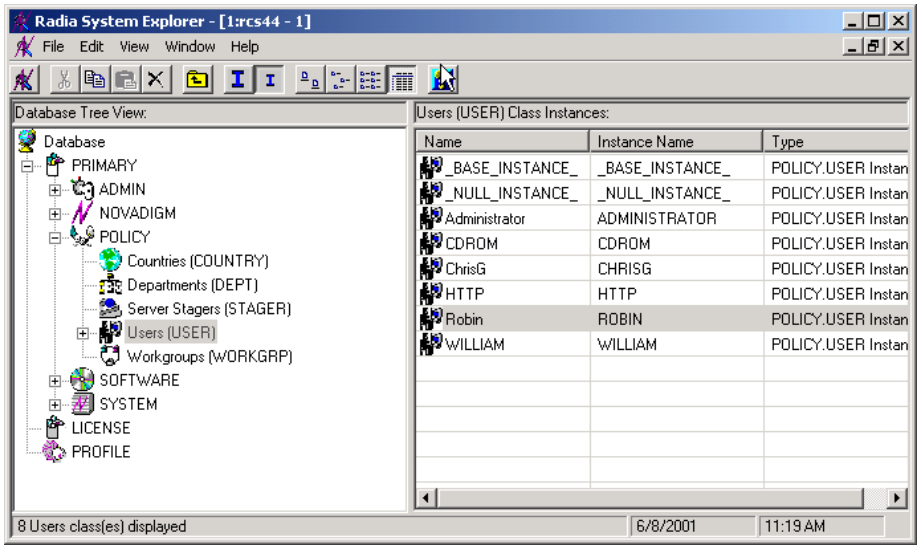
130 Select **New Instance**.

131 In the Create Instance dialog box, type a display name (up to 25 characters) and instance name (up to 25 characters).



132 Click **OK**.

The user instance, Robin, is created.



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 System Explorer to assign the user Robin to the Sales department.

The Sales instance, shown in the Departments (DEPT) class in these examples, 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 168. However, instead of right-clicking USER, you would right-click the appropriate class, such as Departments (DEPT).

The following instructions use the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To assign a user to a department

1 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

The System Explorer Security Information dialog box opens.

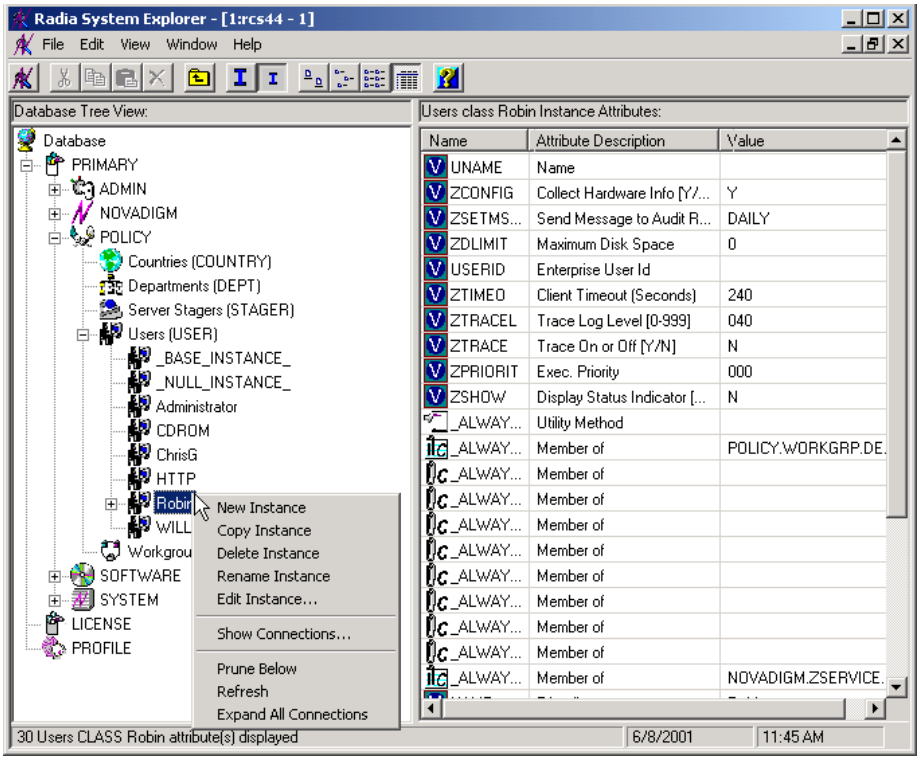
2 If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.

133 Double-click **PRIMARY**.

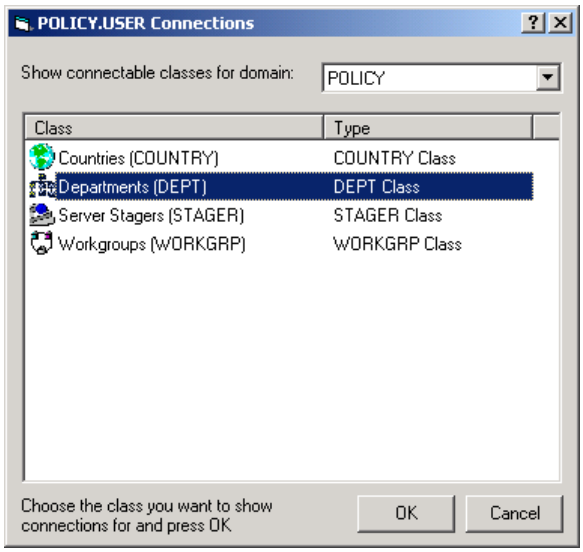
134 Double-click **POLICY**.

135 Double-click **Users (USER)** to open the list of all user instances.

136 Right-click the user instance (in this example, Robin) and select **Show Connections**.

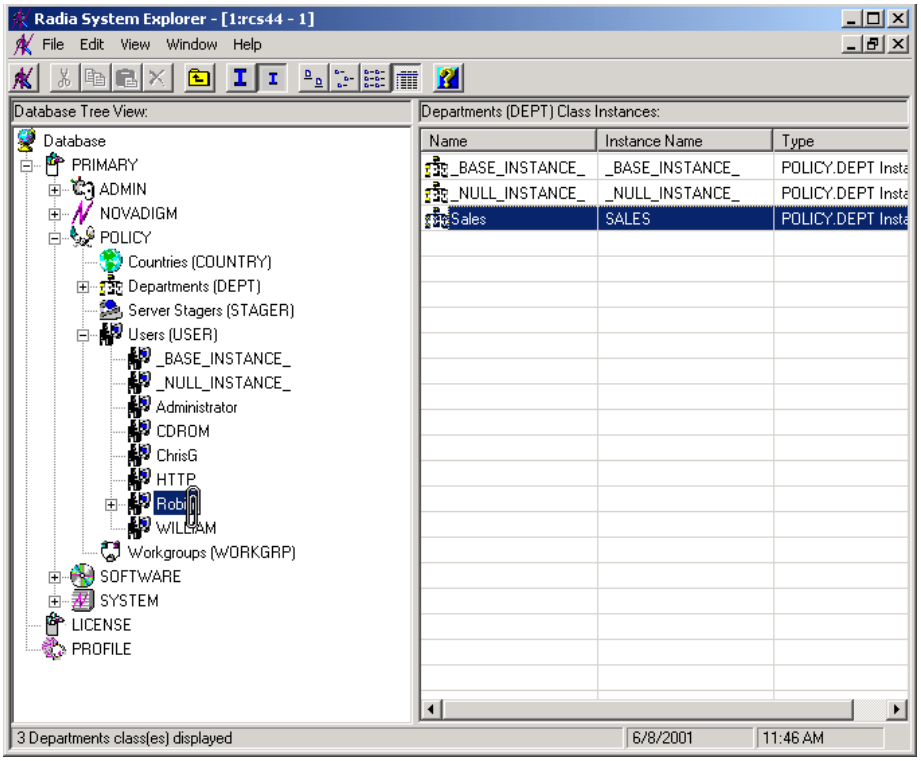


The POLICY.USER Connections dialog box opens. This dialog box displays a list of classes that you can connect the selected instance to.

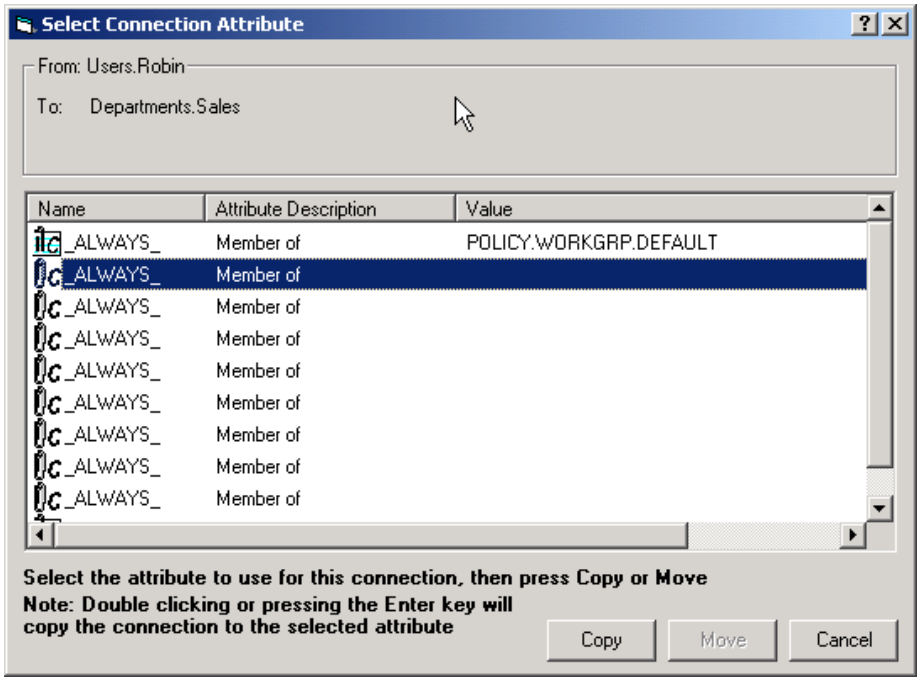


137 Select **Departments (DEPT)** and then click **OK**. The DEPT class instances appear in the list view of the System Explorer. This allows you to easily make a connection between an instance in the DEPT class and an instance in the USER class.

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



The Select Connection Attribute dialog box opens.

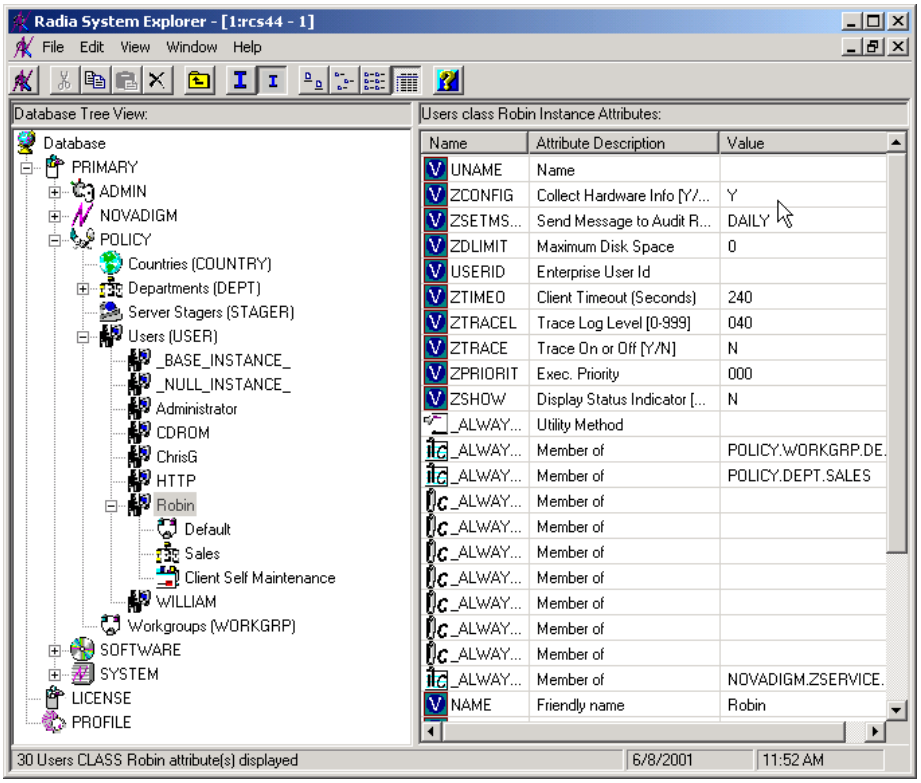


139 Click **Copy** to create the connection from **Users.Robin** to **Department.Sales**.

140 Click **Yes** to confirm the connection.

141 Click **OK** when you receive the confirmation that "Robin has been connected to Sales."

In the System Explorer tree view, notice that **Sales** is now listed under the **Robin** user instance, which indicates that Robin is part of the Sales department.



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.

If you are using the Radia Policy Manager, refer to the *Policy Manager Guide* 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 System Explorer to authorize all subscribers in the Sales department for the Moneydance application.

The following instructions use the System Explorer. The System Explorer is currently available for 32-bit Windows platforms. For more information, refer to the *System Explorer Guide*.

To connect the Moneydance application to the Sales Department

1 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

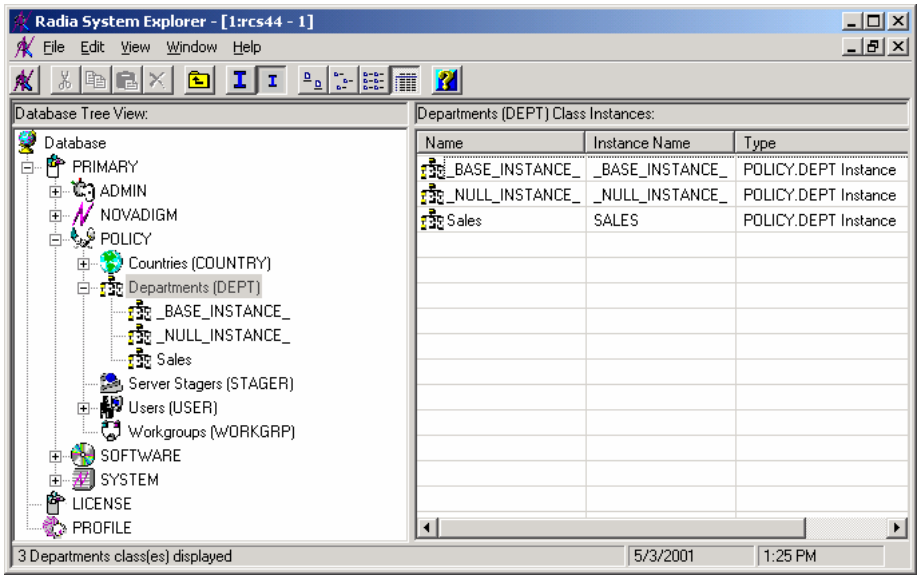
The System Explorer Security Information dialog box opens.

2 If necessary, type a User ID and Password, and then click **OK**. The System Explorer window opens.

142 Double-click **PRIMARY**.

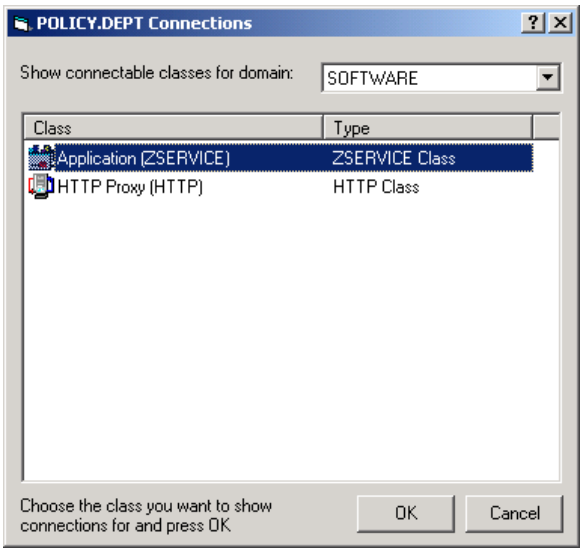
143 Double-click **POLICY**.

144 Double-click **Departments (DEPT)** to open the Departments class.

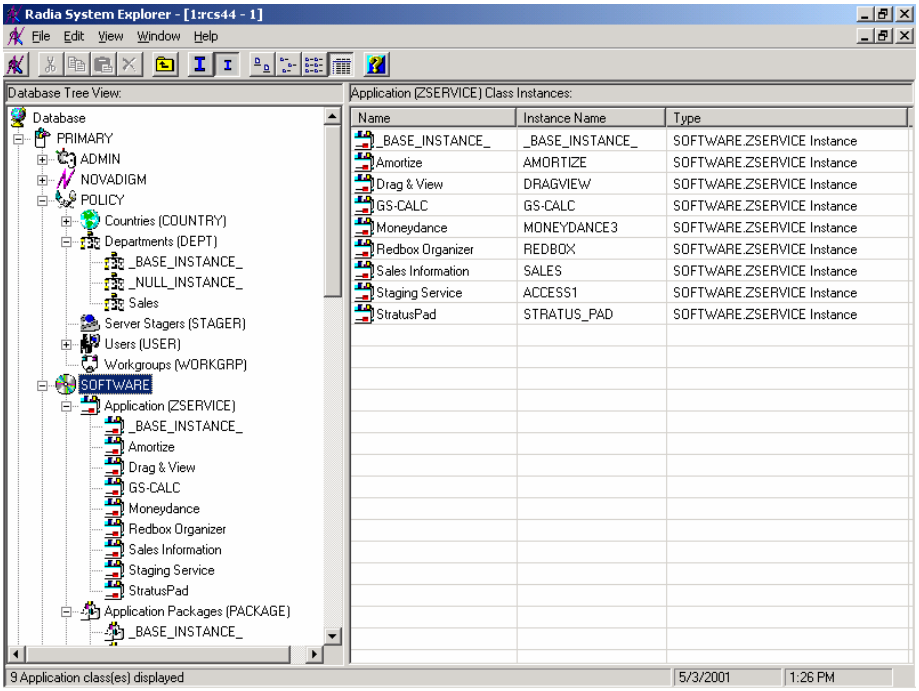


145 Right-click the **Sales** instance (in the tree view), and from the shortcut menu select **Show Connections**. The POLICY.DEPT Connections dialog box opens. This dialog box displays a list of classes that you can connect the selected instance to.

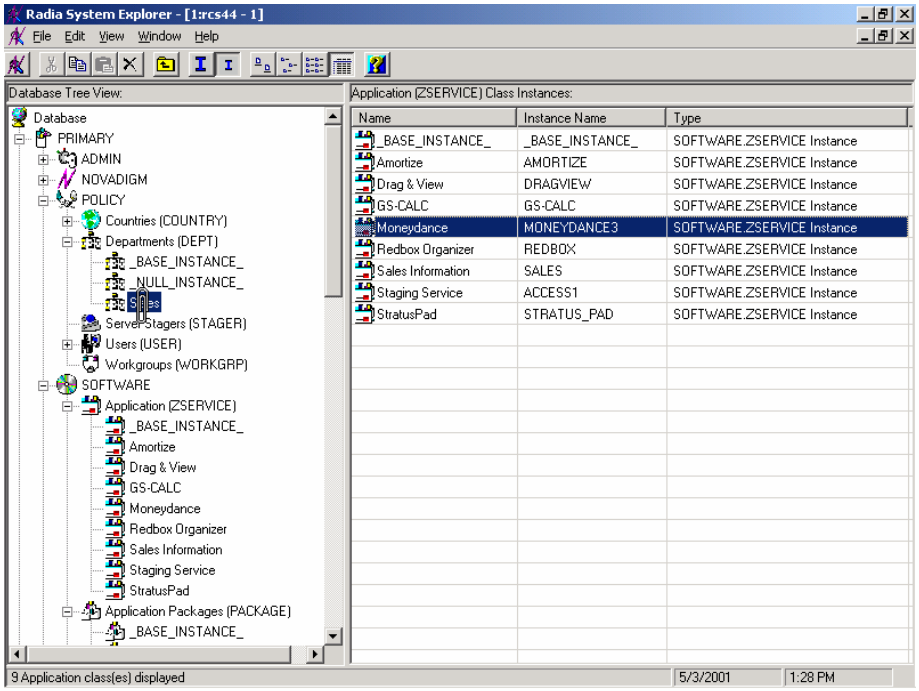
146 From the **Show connectable classes for domain** drop-down list, select **SOFTWARE**.



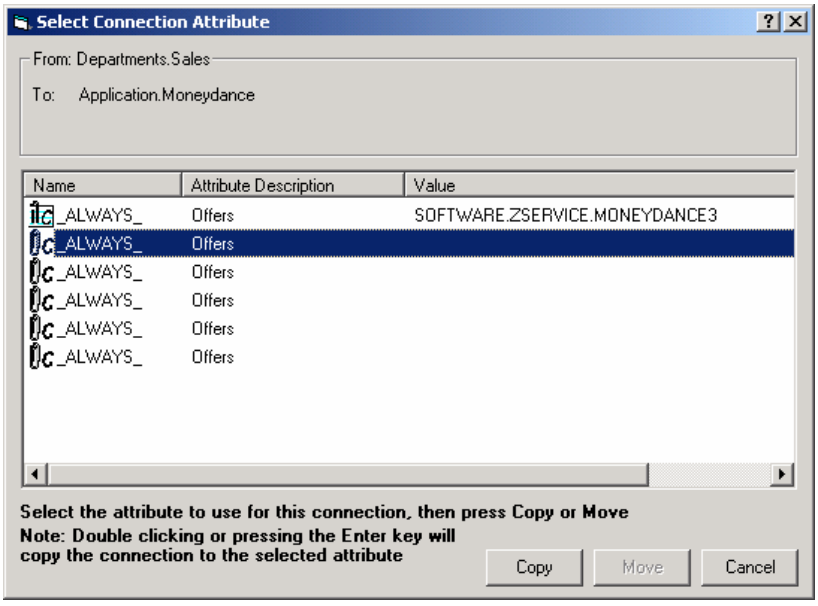
147 Click **Application (ZSERVICE)**, and then click **OK**. The instances in the ZSERVICE class appear in the list view.



148 Select the **Moneydance** 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 as shown below, release the mouse button.



149 The Select Connection Attribute dialog box opens.

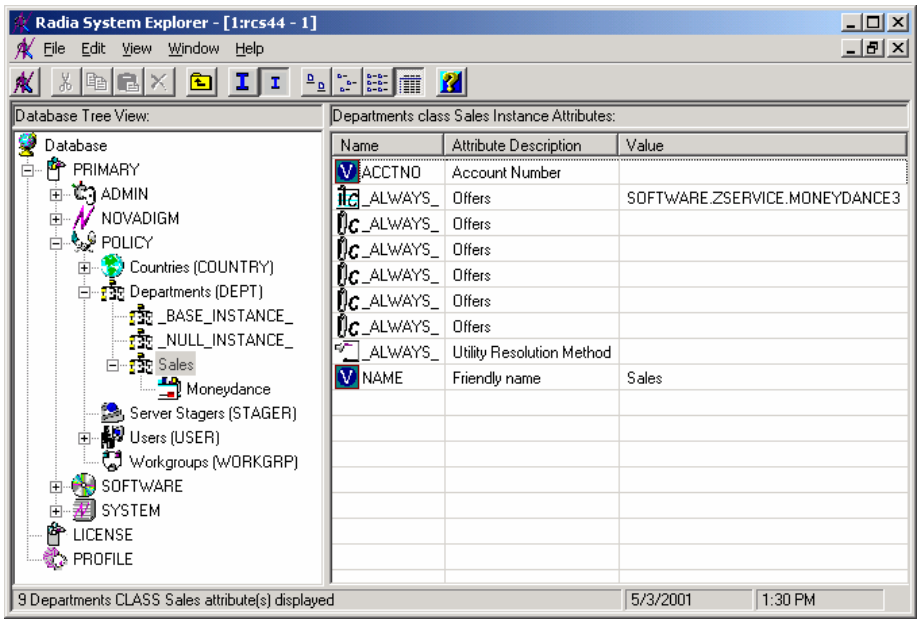


150 Click **Copy** to create the connection from **Departments.Sales** to **Application.Moneydance**.

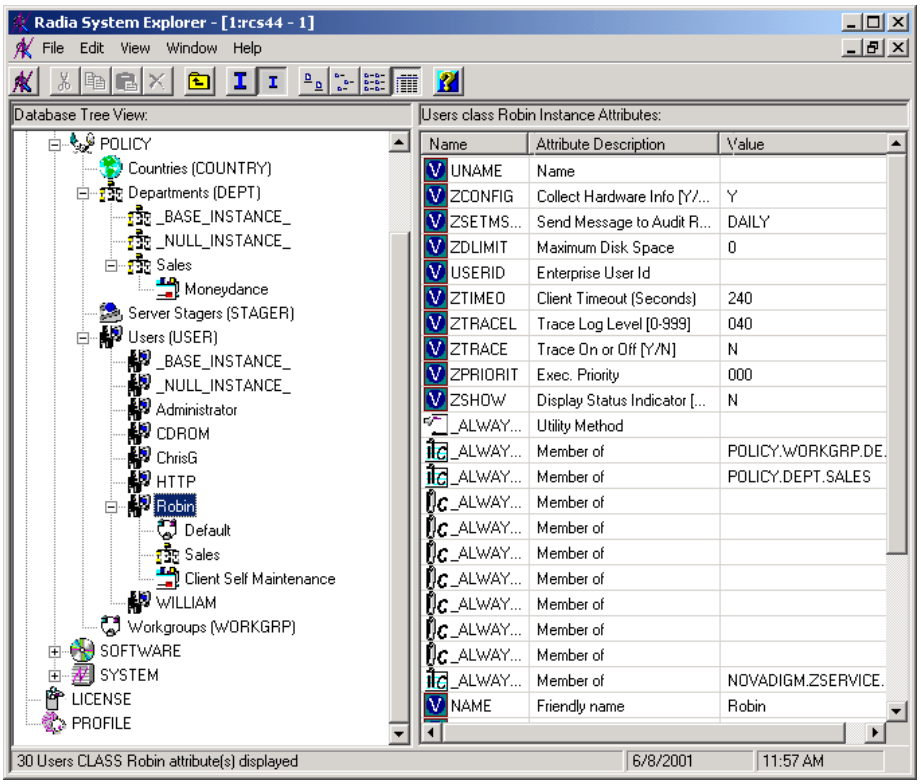
151 Click **Yes** to confirm the connection.

152 Click **OK** when you receive the confirmation that "Sales has been connected to the Moneydance."

In the System Explorer tree view, notice that Moneydance is listed under the Sales department instance, which indicates that the entire Sales department is now authorized to receive the Moneydance application.



Notice now that Robin, listed in the Users (USER) class, is part of the Sales department. You can also see that the Moneydance application has been authorized for the entire Sales department. Therefore, as long as Robin is part of the Sales department, Radia will manage the Moneydance application on his computer.



You can see how using groups simplifies assigning applications to users. You can modify the applications that the individuals in the Sales department are authorized for, simply by manipulating the connections between the applications and the Sales department group. And, you can add users to the Sales department, quickly authorizing them for a series of applications. Or, you can remove users from the Sales department, taking away their authorization to applications.

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.

7 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 *suggested* implementation for the 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 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.

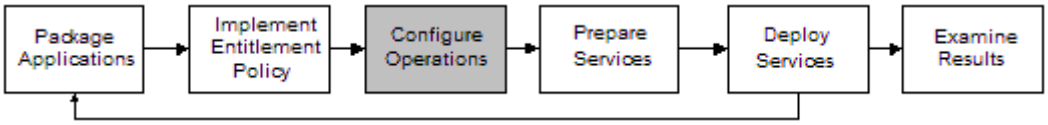


Figure 8: Tasks completed in this guide.

Radia Client Operations Profiles

If you have multiple Configuration Servers, 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 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 Configuration Server to use based on its functional role, allowing for load balancing among Configuration Servers.
- Enhanced diagnostics capabilities.

To use Client Operations Profiles, you must be using the 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 Software Manager User Interface.
- **Server Access Profile (SAP)**
  Use instances in the Server Access Profile (SAP) class to define 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.

153 Set criteria for Server Access Profile instances.

154 Set priorities for Server Access Profile instances.

155 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 Configuration Server, 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 Configuration Server can be identified as type RCS. A Configuration Server, 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 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 Configuration Server to get the client computer's Client Operations Profile.

- **Service resolution (S)**
  Use this Configuration Server to resolve the client computer's services.
- **Client self maintenance (M)**
  Use this Configuration Server to perform Client Self Maintenance.
- **Reporting (R)**
  Use this 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 Configuration Server to download application data to the client computer.
- **All (A)**
  Use this Configuration Server for any of the roles listed above.

A Proxy Server or CD-ROM can only serve the role of Data download (D). A Configuration Server can serve any of the above roles.

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 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 Configuration Server must be either D or A.

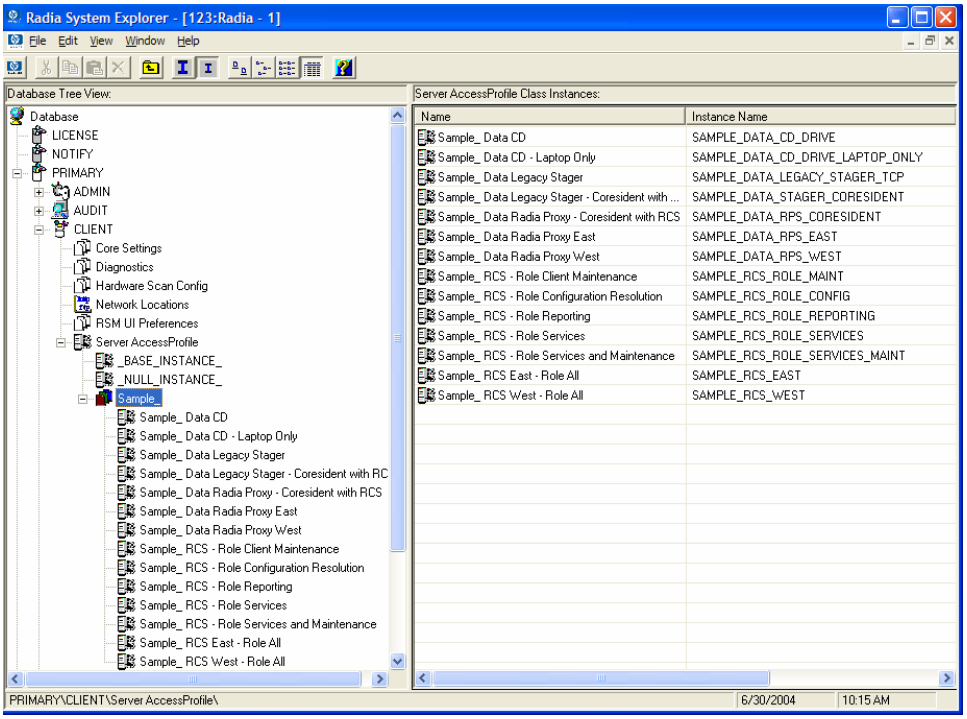
Task 1 Identify Radia Servers

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 Configuration Server, Proxy Server, or CD-ROM. Radia Servers with TYPE=DATA, must have a role of Data download (D). Configuration Servers, set to TYPE=RCS, can serve many roles. You will need to decide which roles your Configuration Servers can perform.

Only a 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.

Task 2 Create Server Access Profile Instances (SAP)

Use System Explorer to create one SAP instance for each Server Access Profile. Table 18 on page 189, describes the attributes in the SAP class. We provide you with samples for each type of Server Access Profile in the Radia Database. The figure below displays the SAP samples we provide.



Use 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 18 on page 189, to configure the instance for your enterprise.

Instances in the SAP class are used before the IP specified in the radskman command line.

Table 18: 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 Configuration Server. Set to Data for 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 Configuration Server.
URI	**Universal Resource Identifier** Create the Universal Resource Identifier to specify the Configuration Server, or Proxy Server. See Table 19 on page 193 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 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.

Attributes	Description
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 Configuration Server if the Configuration Server pushes back on the Client Connect. Set to 1 to 999 for number of retries if the Configuration Server pushes back. Default: The default setting is 0.
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.

Attributes	Description
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: • Application Manager: RAM • Inventory Manager: RIM • 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 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.

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.

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 19 on page 193.

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 19:

Table 19: URI Example

SAP Type	URI and TYPE attributes are set to:
Configuration Server over TCP/IP using default port of 3464	URI = tcp://novarcs:3464 TYPE = RCS
Configuration Server over TCP/IP using port 7800	URI = tcp://novarcs:7800 TYPE = RCS
Configuration Server using SSL on port 443	URI = tcps://novarcsssl:443 TYPE = RCS
Proxy Server using HTTP	URI = http://novarps:3466 TYPE = DATA
CD-ROM	URI = file://&(ZCONFIG.ZHDWCDDR)|/DATA/ TYPE = DATA

Task 11 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 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 ZNETLOC. The ZCONFIG.ZNETLOC 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. Enable Client Operations Profiles on page 198 shows you how to connect the subscribers to the appropriate location class using the client computer's ZCONFIG.ZNETLOC variable.

Task 12 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. The figure 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.

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.

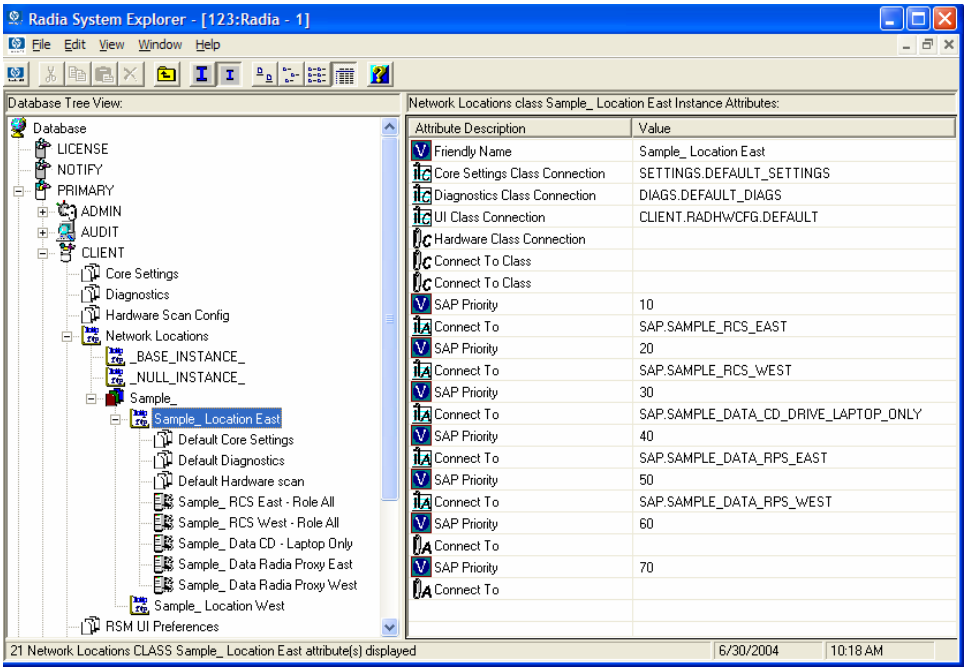


Table 20 on page 195 describes the attributes of an instance in the LOCATION class. For example, if you want your SAMPLE_RCS_ WEST to be used before your SAMPLE_RCS_ EAST, increase its priority to 7. To do this, change the SAPPRI for SAMPLE_RCS_WEST to 7 as shown in the figure below.

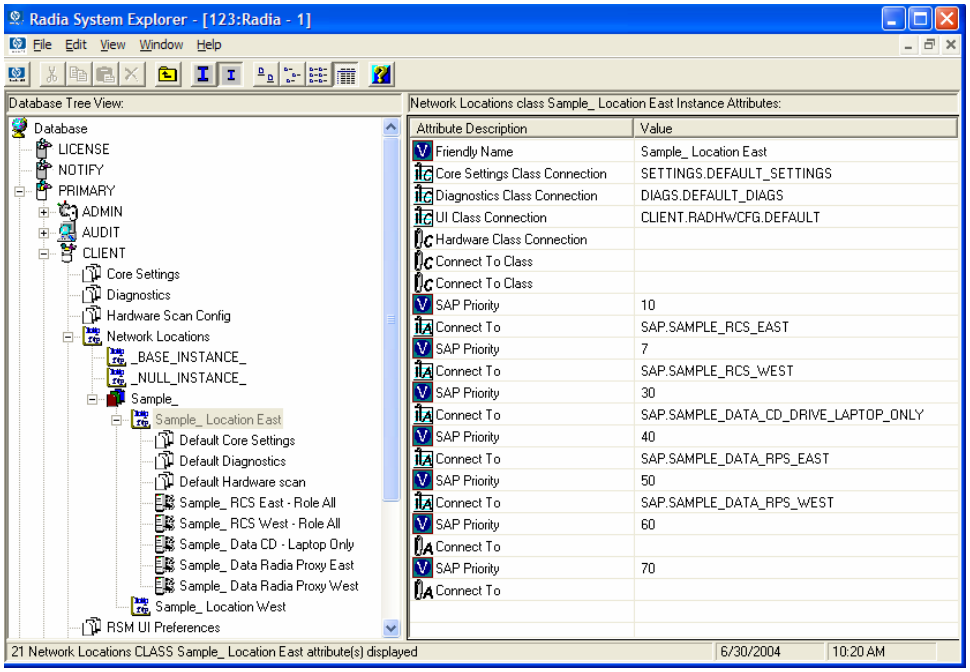


Table 20: Attributes of the LOCATION Class

Attribute	Description
NAME	**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.
ALWAYS	**UI Class Connection** Specify an instance in the RADUICFG class.
ALWAYS	**Hardware Class Connection** Specify an instance in the RADHWCFG class.

Attribute	Description
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.

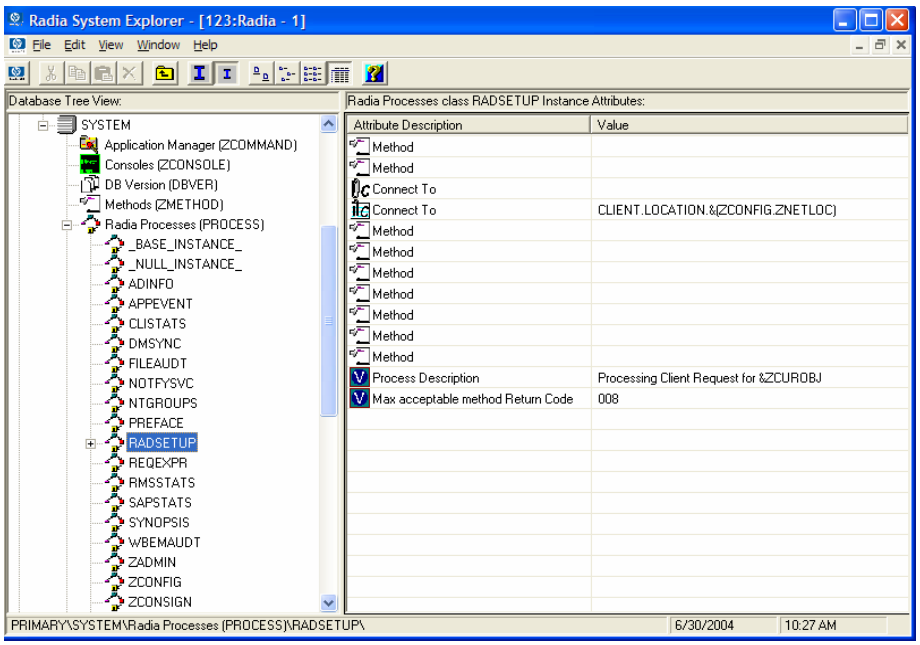
Attribute	Description
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 40.
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. 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.

Task 13 Enable Client Operations Profiles

There are two phases to this step. First, you will need to create a process on the 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 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. The next figure shows an example.



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 objects.txt to add the COP variable to the RADSETUP object. To do this, add the following to the objects.txt file:

  ```
  RADSETUP COP = Y
  ```

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`.

  Add the following lines to the `initmeth.rex`:

  ```
  /* Sample INITMETH.REX to Enable the COP */

  call edmget('RADSETUP',0)

  RADSETUP.COP = 'Y'

  call edmset 'RADSETUP'
  ```

- 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.

If similar attributes exist in both the SETTINGS and SAP classes, the attribute in the SAP class will be used.

Table 21: Attributes of the SETTINGS Class

Attributes	Description
NAME	**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 Configuration Server if the Configuration Server pushes back on the Client Connect. Set to 1 to 999 for number of retries if the Configuration Server pushes back. Default: The default setting is 0.

Attributes	Description
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.
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 Configuration Servers, set RCSDATA to N. Default: The default setting is Y.
ADINFO	(Windows Only) **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 Configuration Server for all resolution processes. Default: The default setting is Y.

Attributes	Description
ZGRPINFO	(Windows Only) **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 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 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 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.
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.

Attributes	Description
ALWAYSS	**Always Upload CFG Objects [Y/N]** Set to Y to always upload all objects in the RADSETUP directory to the Configuration Server. The default location for the RADSETUP directory is `/opt/Novadigm/lib/`. 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 `/opt/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 `/opt/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 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 `/opt/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.

Attributes	Description
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.
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	(Windows Only) **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.

Attributes	Description
USEDEFS	**Use Default SAP [Y/N]** If a SAP cannot be found for the needed ROLE, specify Y to default to the Configuration Server set on the command line.
DEFROLE	**Default SAP ROLE (A,O,S,M,R)** Specify roles for the Configuration Server specified on the command line. If not specified, the ROLE is set to A (All), and the Configuration Server will be able to perform any ROLE. Note: USEDEFS must be set to Y to use DEFROLE.
RAD2XUI	**Enable RADUI 2.x** Default setting Y displays terminal messages. Set to N to turn off UNIX console messages.
RSTROPT	(Windows only) **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 100 KB, 86 KB, 10 MB. First the client calculates the network bandwidth. One of two scenarios will apply: If the network bandwidth is under 86 KB, the file size is compared to 100 KB. If the file size is over 100 KB, checkpoint restart is enabled for that file. If the network bandwidth is over 86 KB, the file size is compared to 10 MB. If the file size is over 10 MB, 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.

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	(Windows Only) **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. Software Manager subscribers will receive a dialog box showing just an OK button that RSM needs to be updated. 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 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 Software Manager Interface. P is the same as I for Application Manager users. Default: I

Attributes	Description
SENDRPT	**Send Reporting Object [I/D]** Set to D to defer sending all reporting objects to 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 Configuration Server immediately (I) after they are created. This requires multiple disconnects and reconnects to the 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 "Clie

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 22: Attributes of the DIAGS Class

Attribute	Description
NAME	**Friendly Name** Type the friendly name of the instance.

Attribute	Description
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.

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.

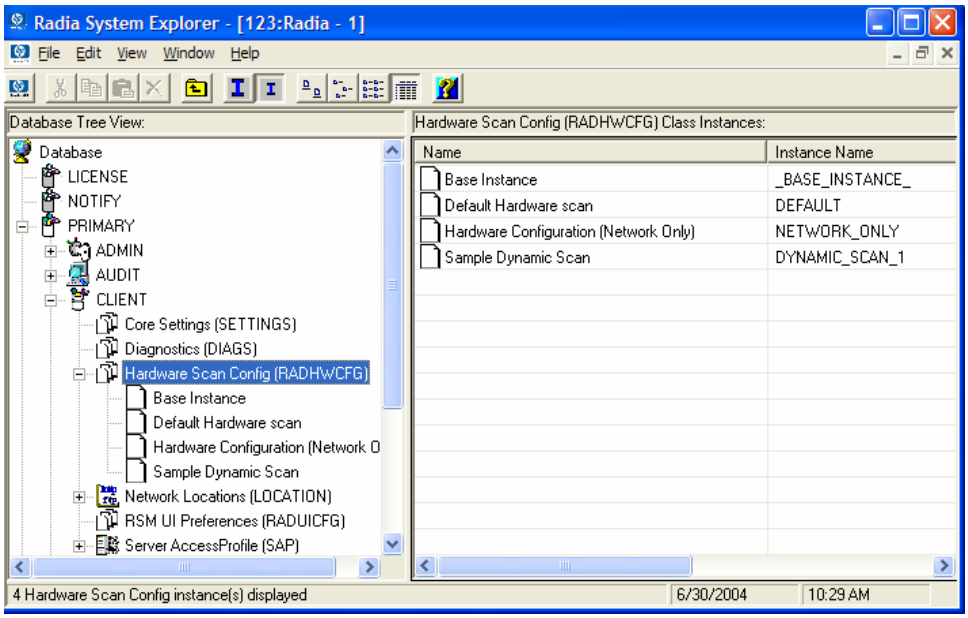


Figure 9: View the sample instances in the RADHWCFG class.

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 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 ZHDWPA0*n* attribute will be reported in ZCONFIG.

Table 23: 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

Attribute	Description
HDREMOTE	(Windows Only) **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	(Windows Only) **Peripherals [Y/N]** Specify Y to scan for peripherals such as keyboard and mouse. ZCONFIG attributes: ZHDWKYBD, ZHDWMOUS, ZHDWPPAR, ZHDWPSER, ZHDWVIDO, ZHDWVRES
PRINTER	(Windows Only) **Printers [Y/N]** Specify Y to scan for printers. ZCONFIG attributes: ZHDWPA00, ZHDWPA01, ZHDWPPRN
HAL_VER	(Windows Only) **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	(Windows Only) **Application Version [Y/N]** Specify Y to scan for versions of MSI (ZHDWVMSI) and IE (ZHDWVIE).

Attribute	Description
WMISCAN	(Windows Only) **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. File is the only type of dynamic scan instance supported for the UNIX version of Radia. 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 File, and Parm*n* is the query for the information.

Example: 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 `/opt/temp/test`, create a DSCAN002 similar to:

```
TEST####=FILE(/opt/temp/test;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 `/opt/temp/test` 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 Software Manager User Interface. You must be licensed for the Software Manager client. To implement the hardware scan options, connect an instance of the RADUICFG

class to an instance in the LOCATION class. For a description of the RADUICFG class and instances, refer to the *Software Manager Guide*.

You must be licensed for and install the Software Manager to use this class.

Client Operations Profile Example

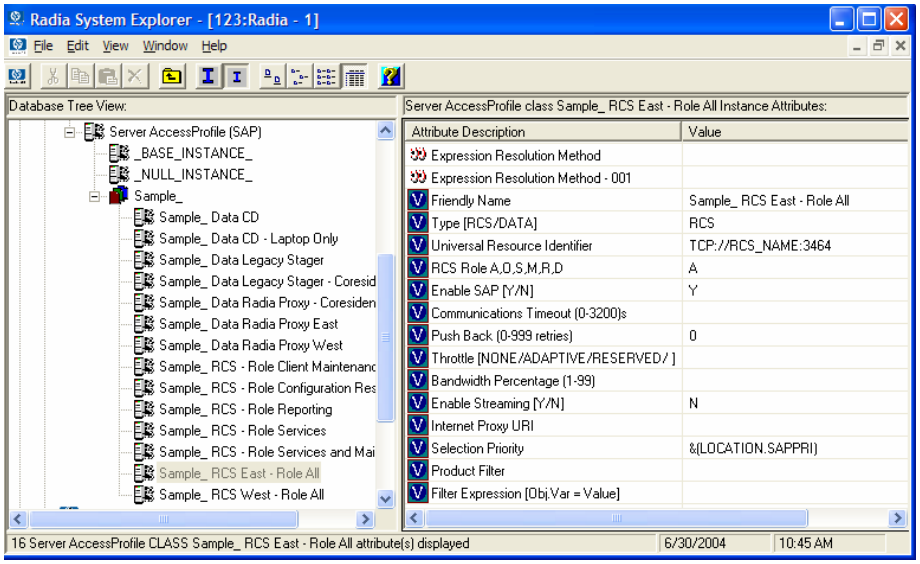
This section provides a simple example of how to configure a Client Operations Profile. The driving force is to have your client computers connect with the most appropriate Configuration Server. Usually, you will want to assign your client computers to a Configuration Server based on network address.

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 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. The figure below shows a SAMPLE_RCS_EAST.



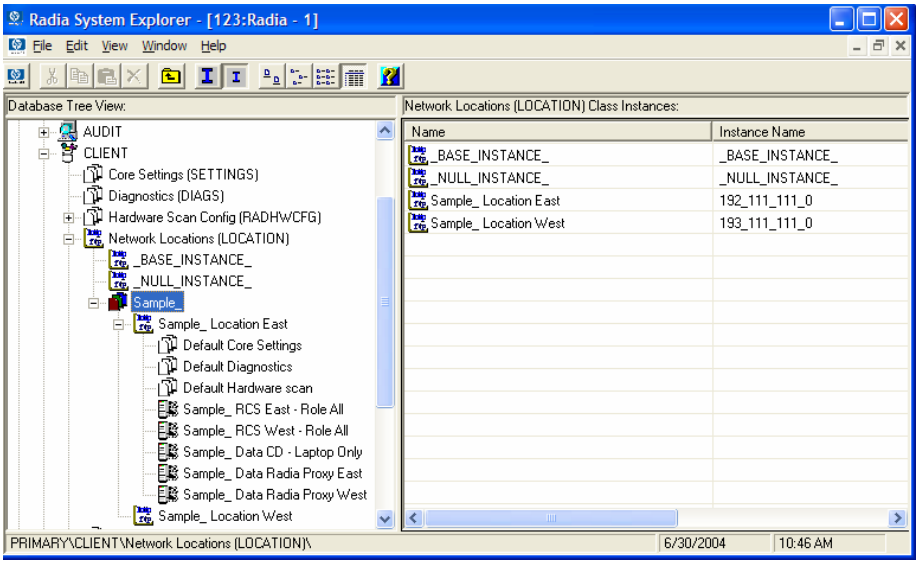
2 In the Server Access Profile (SAP) class use the TYPE attribute to specify a server as type DATA or RCS. In this example, we will be configuring only Configuration Servers. Therefore, all servers will have SAP.TYPE set to RCS.

156 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 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.

157 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.



158 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.

159 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 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.

8 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 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 Application Manager. This chapter covers preparing services.

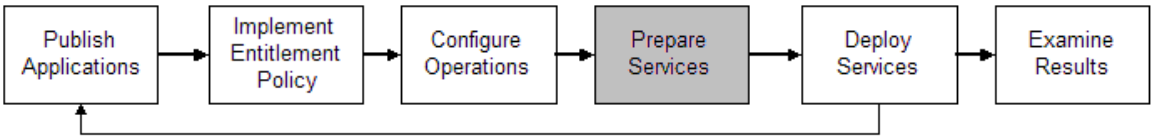


Figure 10: Tasks completed in this guide.

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.

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 24 on page 221 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.

The parameters for the reboot attribute are not case-sensitive

Table 24: 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 Software Manager and Application Manager. 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.

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: 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.

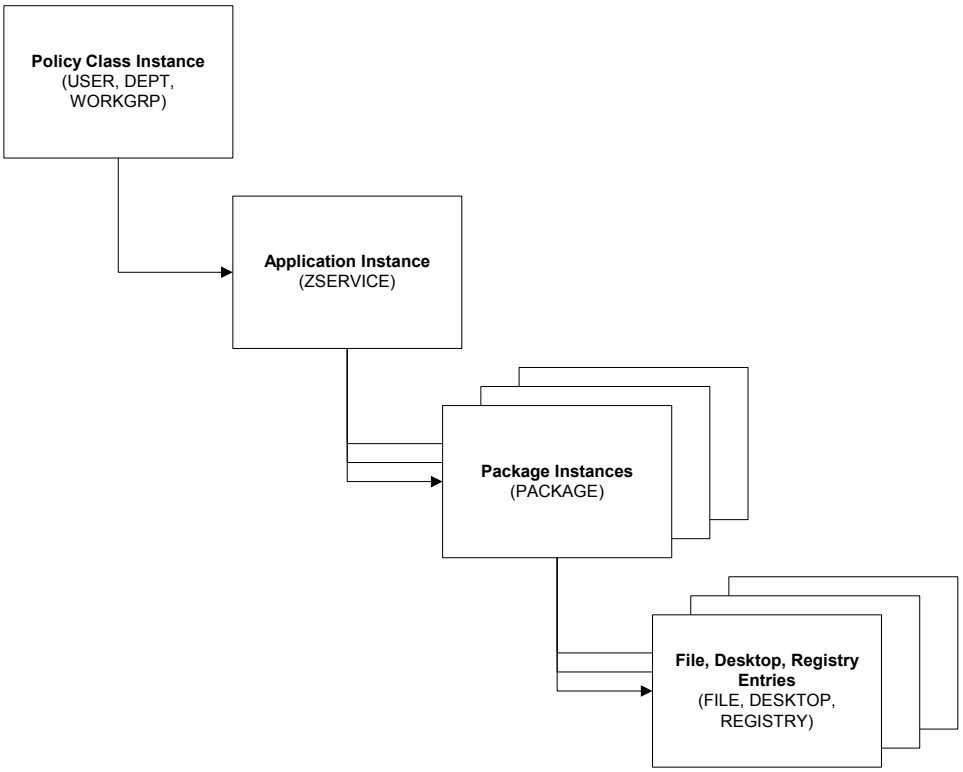


Figure 11: Model for non-versioned deployments.

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.

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.

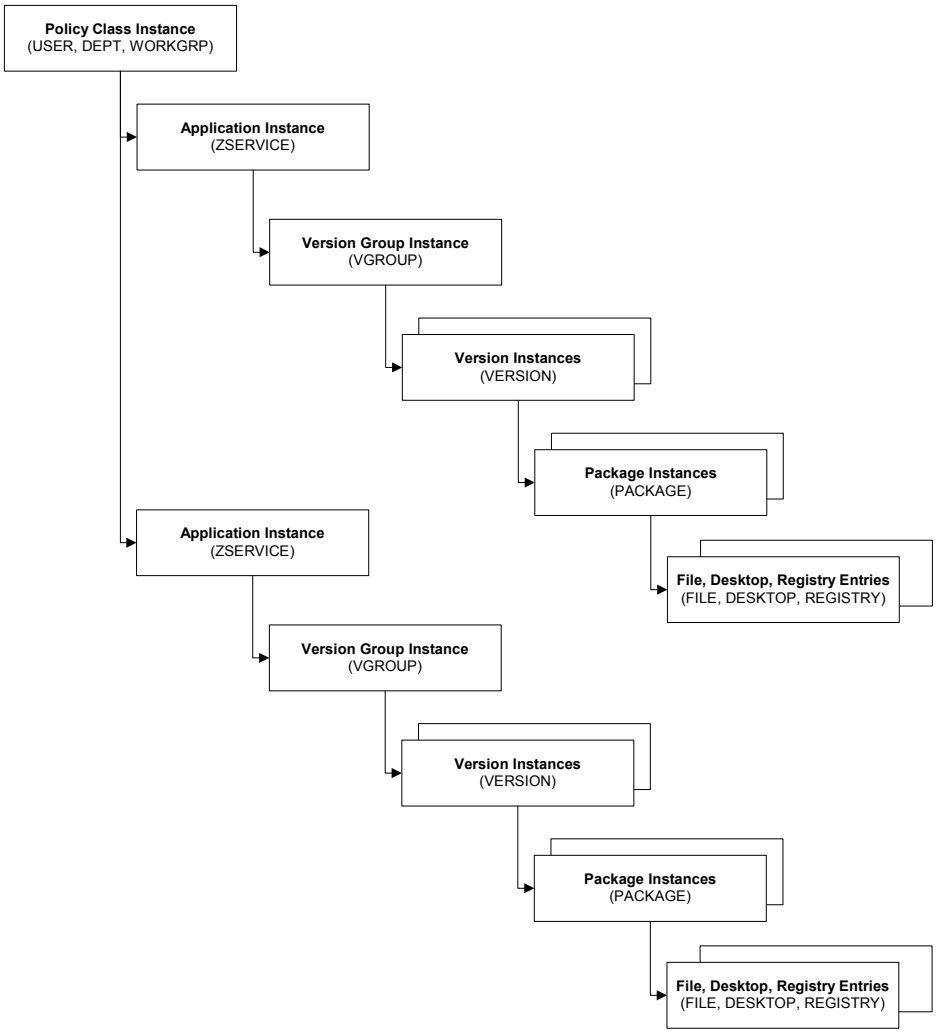


Figure 12: Model for versioned deployment.

To prepare versioned applications

1 Use the 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 below to finish creating the Version Group.

The Version Group Editor

Use the Version Group Editor, in the 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 System Explorer to create a new instance in the Version Group (VGROUP) class.

To create a Version Group

1 Go to **Start** → **Programs** → **Radia Administrator Workstation** → **Radia System Explorer**.

The Radia System Explorer Security Information dialog box opens.

The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This 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.

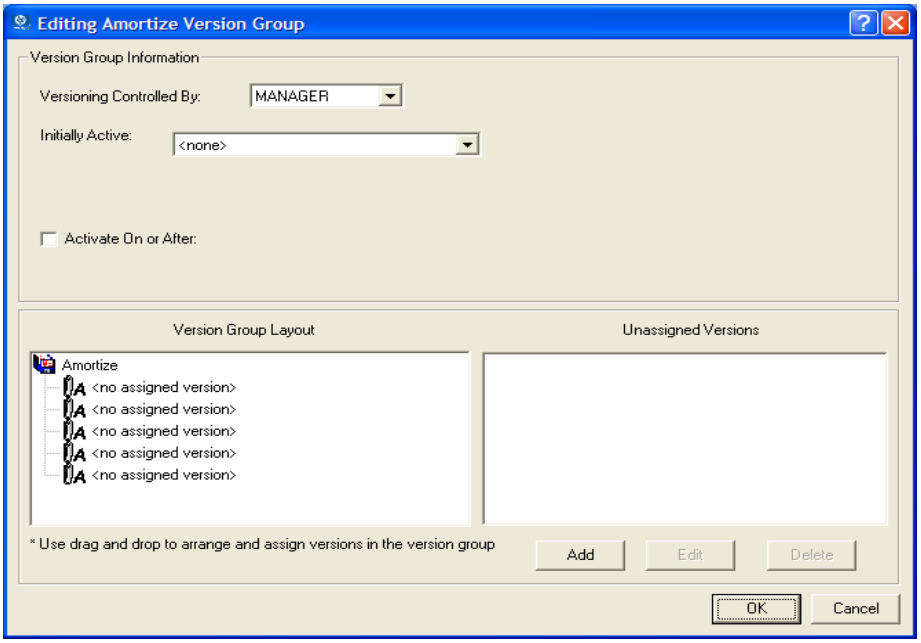
5 Right-click Version Group (VGROUP).

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.

8 Click **OK**.

The Editing Version Group dialog box opens.



Creating a Version Instance

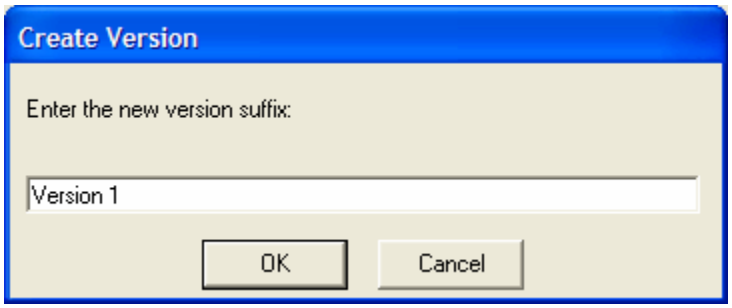
Now that you have created a Version Group (VGROUP) instance, you will learn how to create an instance for each version of your application.

To create a version instance

1 In the Version Group Editor, click **Add**.

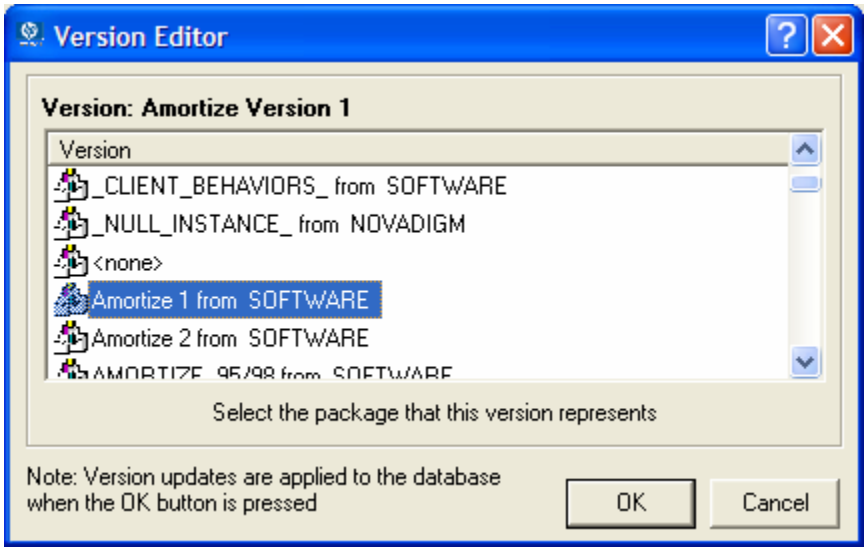
The Create Version dialog box opens.

2 Type a suffix that identifies the version. For example, type **`Version 1`**. The Version instance will be named Amortize Version 1.



3 Click **OK**.

The Version Editor dialog box opens.

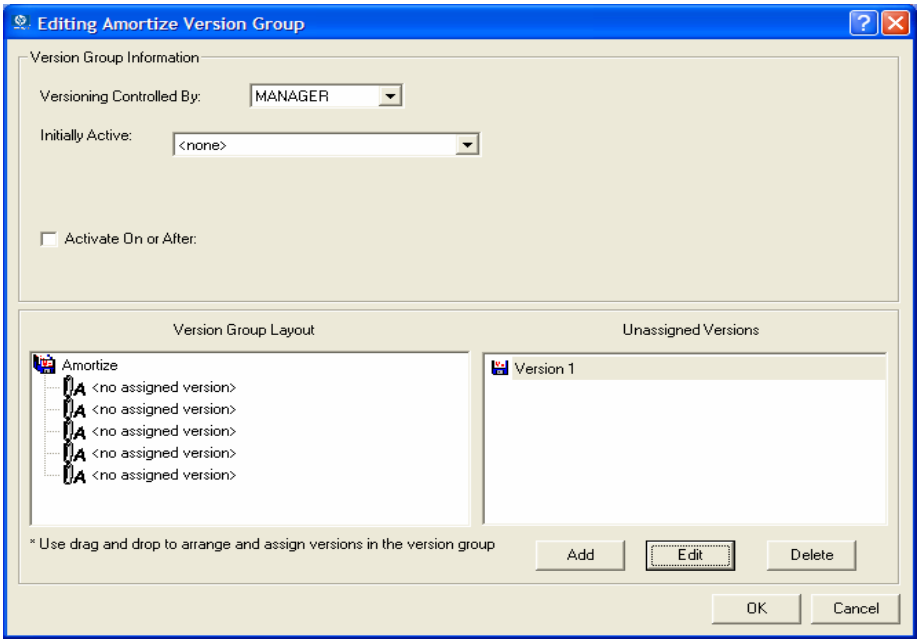


The Version Editor dialog box contains a list of Application Package (PACKAGE) instances stored in the Radia Database. Use this dialog box to connect the new Version (VERSION) instance to an Application Package (PACKAGE) instance. There is a one-to-one correspondence between these two instances.

4 Click the appropriate Application Package (PACKAGE) instance, such as Amortize1.

5 Click **OK**.

The Version instance appears in the **Unassigned Version** list.



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.

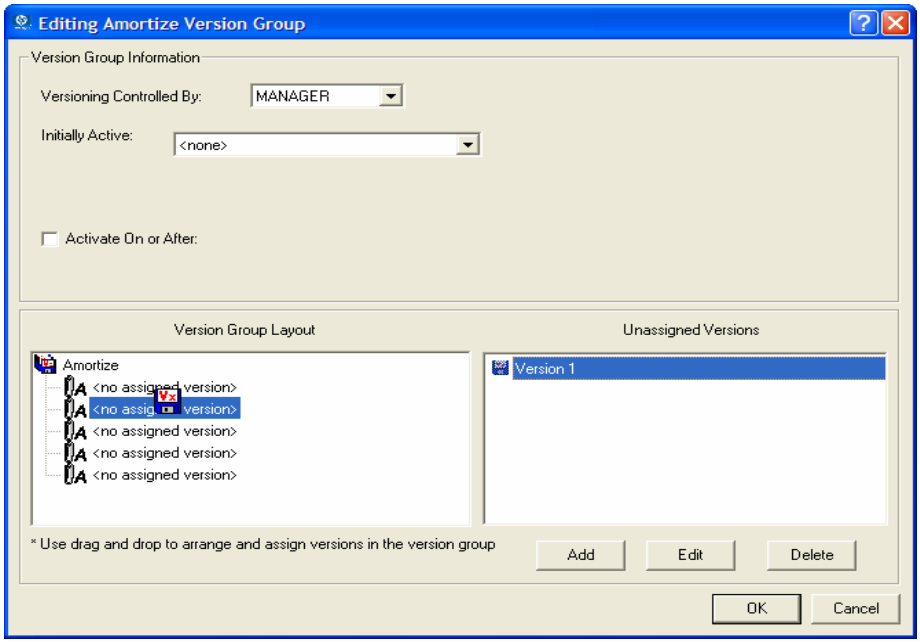
3 Click **OK** to close the Version Group Editor.

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.



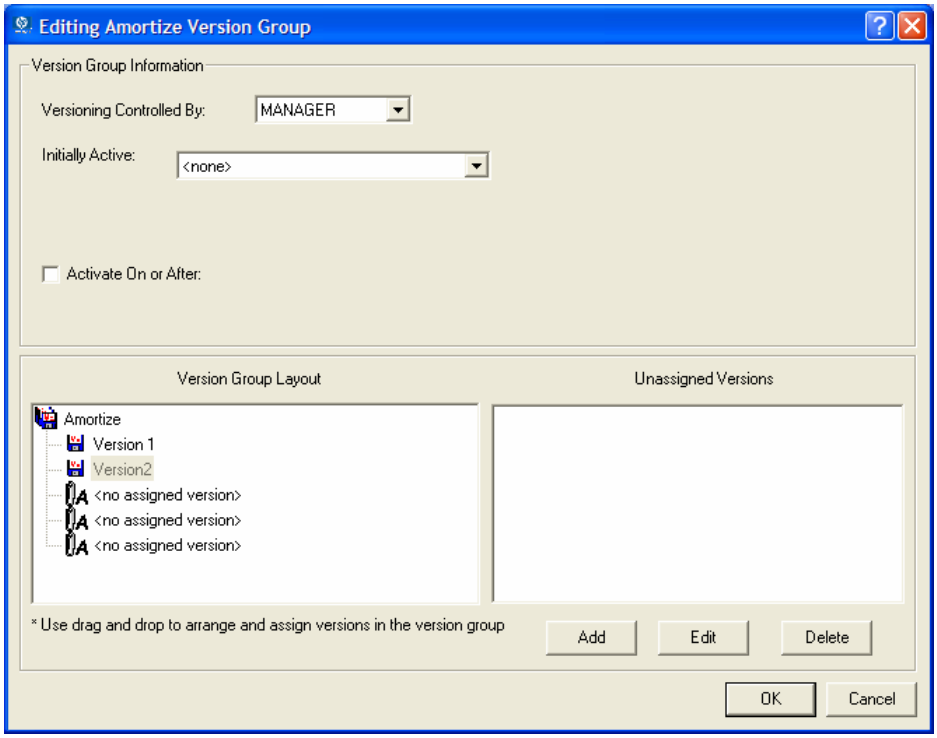
Assign each of the Version instances that you created to the Version Group.

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.



To prepare a Version Group for deployment

1 In the **Versioning Controlled By** drop-down list, select **Manager** or **Client**.

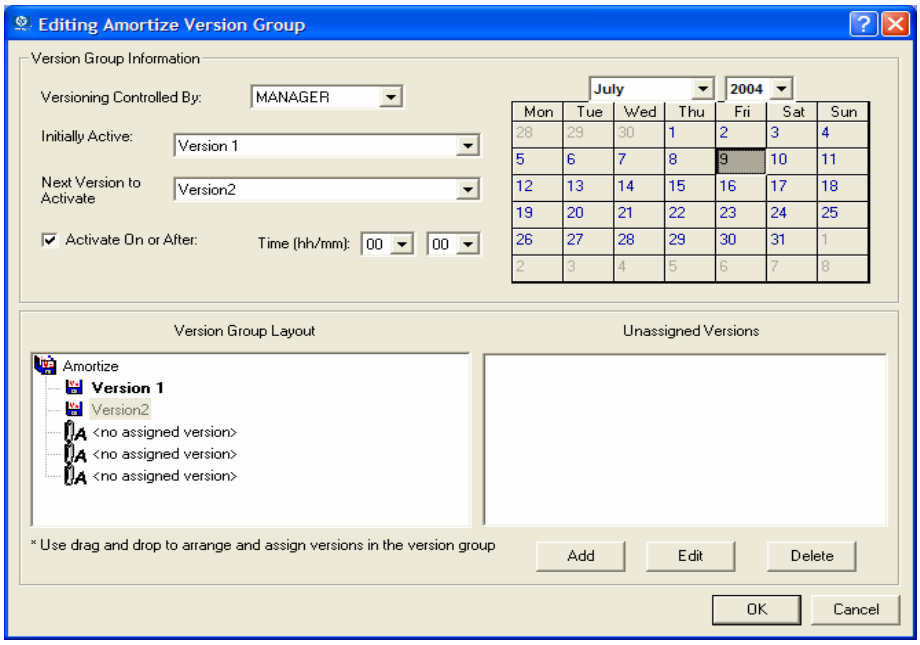
- Select **Manager** if you want to control the version to be deployed.
- Select **Client** if you want the subscriber to control the version to be deployed. *This is only used with the Radia Software Manager client.*

> 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 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 the next figure.

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.

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 Configuration Server knows when to activate the next version.



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.

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 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 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**.

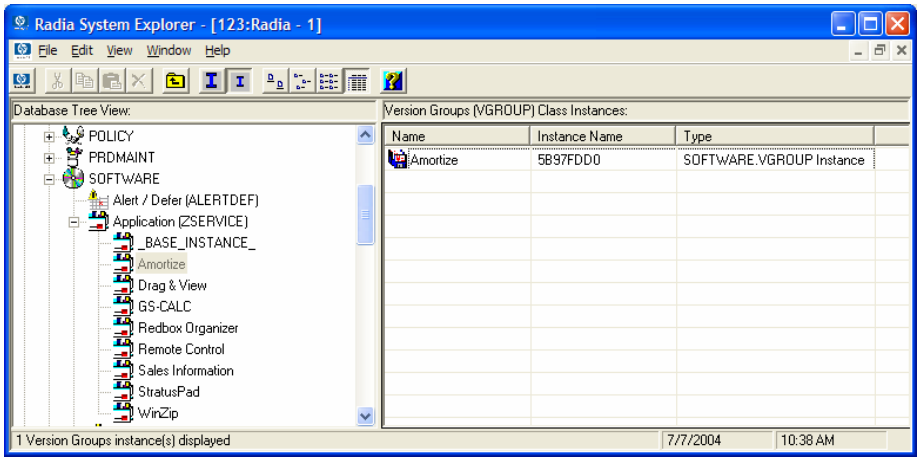
3 Select **Show Connections**.

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 System Explorer.



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.

> 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 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 System Explorer.

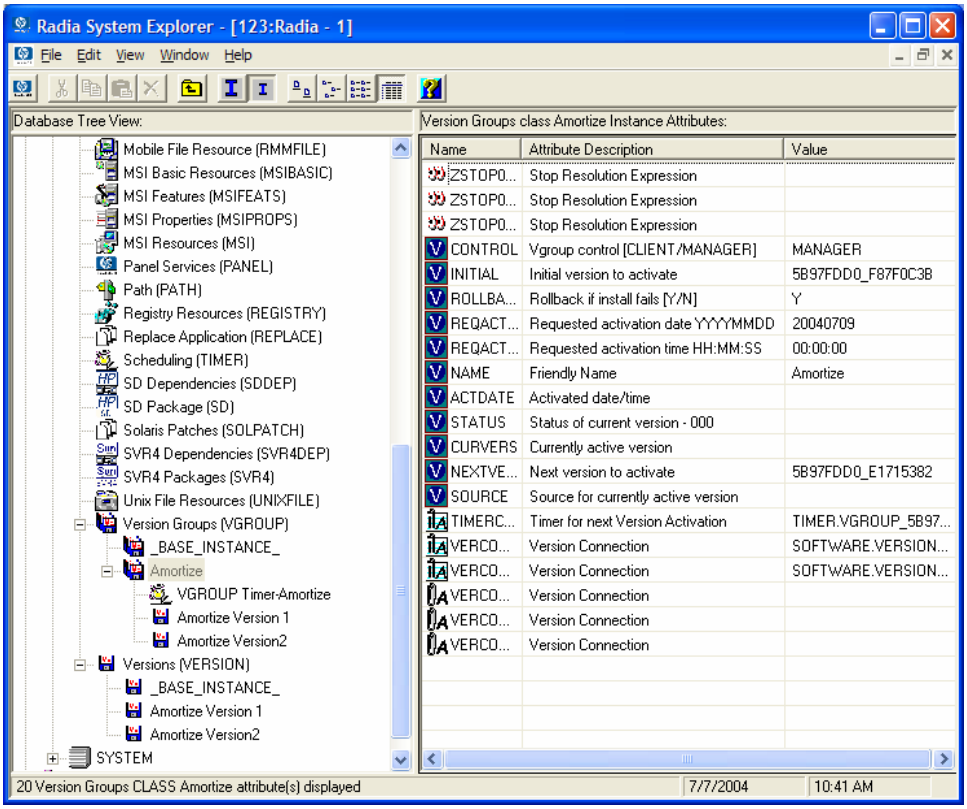


Figure 13: Version Groups (VGROUP) class instance.

The following table describes each of the attributes in the Version Group (VGROUP) class.

Table 25: Version Group (VGROUP) Class Attributes

Attribute	Description
ZSTOP00*n*	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 System Explorer to set this attribute.

Attribute	Description
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 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.

Attribute	Description
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.
VERCON0*n*	Connects to each version in the version group. Each VERCON0*n* 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.

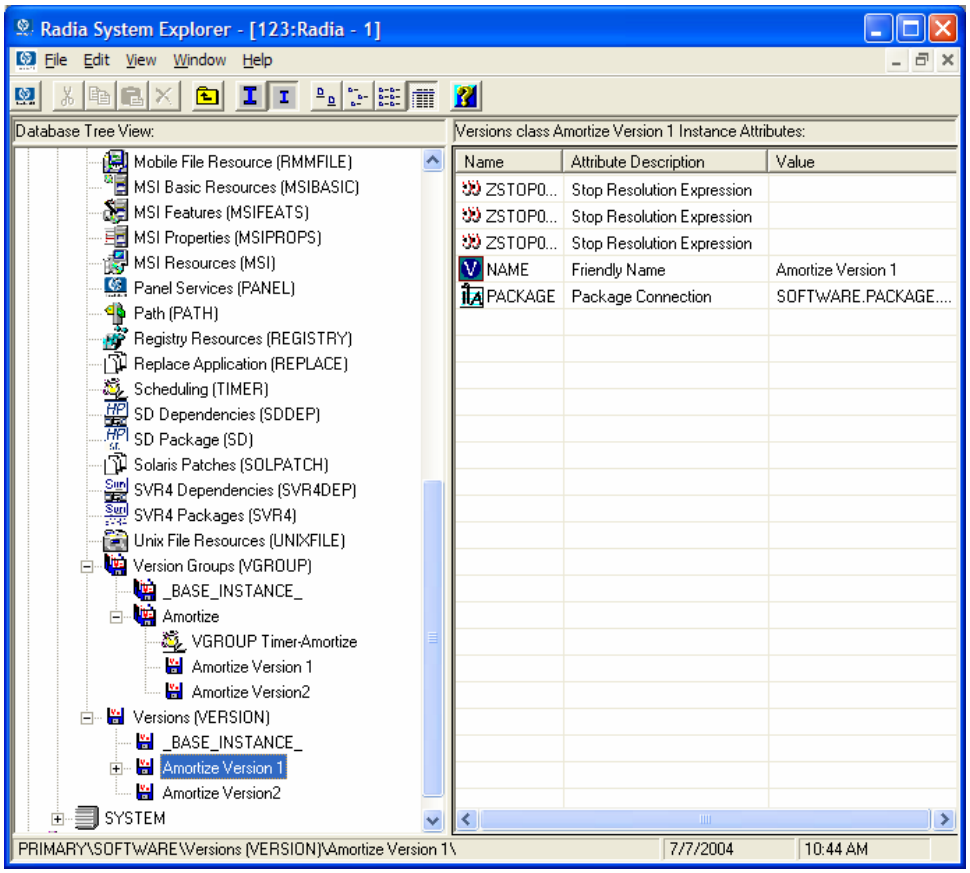


Figure 14: Versions (VERSION) class instance.

The following table describes each of the attributes in the Versions (VERSION) class.

Table 26: Versions (VERSION) Class Attributes

Attribute	Description
ZSTOP00*n*	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 System Explorer to set this attribute.

Attribute	Description
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 System Explorer. Many of the values for these attributes are set when using the Administrator Workstation, such as the Radia Packager or the New Application Wizard in the System Explorer. You can also use the 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 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 27: Modifiable SOFTWARE.ZSERVICE Attributes

Attribute	Description
BDELETE	**Service Pre-Delete Method.** Run before client operations profile resolution.
BREPAIR	**Service Pre-Repair Method.** Run before client operations profile resolution.
BUPDATE	**Service Pre-Update Method.** Run before client operations profile resolution.

Attribute	Description
ZSTOPnnn	**Expression Resolution Method** Stops resolution if the expression evaluates to TRUE. Example: WORDPOS(EDMGETV(ZMASTER,ZOS),'UNIXHPUX UNIXLNUX')=0 This example expression will stop resolution on the instance if the client computer's operating system if the operating system is *not* HP-UX or Linux. In other words, the application will not be installed unless the client computer is running HP-UX or Linux.
ZSVCNAME	**Service Name/Description** Name of the service used for display in the 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, Application Manager or Software Manager. Value is set initially in the New Application Wizard. Possible values are A for Application Manager and S for 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 Application Manager, services are marked as mandatory, M. When using the Software Manager, services are usually marked as optional, O. If you are using Application Manager *and* 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 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

Attribute	Description
	System Explorer Guide for more information on editing a class template. To process mandatory applications using 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.
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.*

Attribute	Description
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 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 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 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 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 Software Manager user interface.
SCHEDOK	**Update Schedule Locally [Y/N]** *For Software Manager only.* Specify Y to allow the subscriber to change the update schedule locally. Specify **N** to maintain control on the Configuration Server.

Attribute	Description
VERSION	**Version Description** Version of the software. This appears in the properties for the service in Software Manager user interface. The value is set initially in the Version field in the New Application Wizard.
NAME	**Friendly Name** This name appears in the properties for the service in the 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:**

Attribute	Description
	• **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.
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 Configuration Server to get new or different components. Specify Y for Yes and N for No.

Attribute	Description
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 applications (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
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 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.

Attribute	Description
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]** *For Windows Installer applications only.* 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	**Percent 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.

Attribute	Description
ZDISCONN	**Disconnect on Install [Y/N]** Allows the client to disconnect from the Configuration Server if there is an open session with the Configuration Server. • Specify Y to disconnect the client from the Configuration Server. • Specify N to keep the client connected to the client from the Configuration Server. Default: N
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).

Attribute	Description
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 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 System Explorer.

Table 28: 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 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 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 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 Configuration Server. The possible values for each are: Y = Yes N = No X = Unknown

Attribute	Description
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 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 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
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 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.

9 Deploying Services

At the end of this chapter, you will:

- Understand the different deployment methods available in Radia and when to use each one.
- Be able to deploy a service at a predetermined time using the Scheduler.
- Know how to use the Notify function to update an application, remove an application, or send an e-mail message to a subscriber.
- Be familiar with key special case deployments.

This guide covers the *suggested* implementation for the 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 Application Manager. This chapter covers the process of deploying applications.

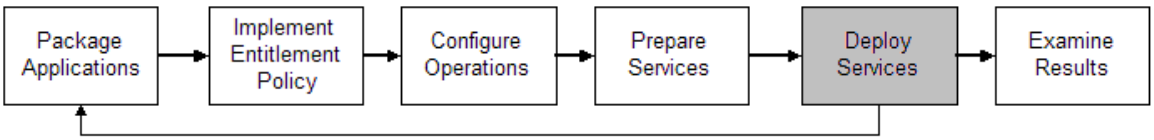


Figure 15: Tasks completed in this guide.

About Deployment Methods

After creating a service using the System Explorer, and deciding which users or groups will receive the application, you are now ready to deploy the service to your subscribers.

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 Configuration Server to install, update, or remove an application, or sends an e-mail to the subscribers of a particular service.

  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.

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 Application Manager and RSM stands for 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 Configuration Server, and how to identify the client computer for policy.

Table 29: 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.

Parameter	Explanation
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 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, /opt/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 Configuration Server. For example, mname=RADSVR01. RAM default: Radia (defaults to RADSTAGE for preload). RSM default: Radia (defaults to RADSTAGE for preload).
port	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, /opt/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.

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). Software Manager does not pass a context by default.

Parameter	Explanation
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, `/opt/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 30: 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
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.

Parameter	Explanation
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 Configuration Server, display of the user logon panel, and when to send objects to the Configuration Server.

Table 31: 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

Parameter	Explanation

context	(Windows Only)
	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 Software Manager. All components are processed.

Parameter	Explanation
flushu	(Windows Only) If you are using `local=y`, set `flushu=y` on user connects (`context=u`) to send reporting objects up to the 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 Configuration Server. Be aware that the user's objects will continue to grow until they are sent to the Configuration Server. On a machine connect (`context=m`), set `flushu=a`, if you want to send all user's reporting objects to the Configuration Server. RAM default: Y RSM default: Y
local	(Windows Only) 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	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	(Windows Only) *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 that is not supported by Radia System Tray. RAM default: N RSM default: Y

Parameter	Explanation
userfreq	(Windows Only) *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 Chapter 7, Configuring Client Operations Profiles.

Table 32: Radskman Client Operations Profiles Parameters

Parameter	Explanation
cop	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

Parameter	Explanation
datauri	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 19 on page 193.
product	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	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 19 on page 193.

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 Proxy Server, and handling SSL security for your Radia clients.

Table 33: 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

Parameter	Explanation
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	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 Configuration Server.
Mnt	Set to Y to process Radia Client Self Maintenance on this connect. Note: Client Self Maintenance will not be deployed unless you set `mnt=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

Parameter	Explanation
Sendcat	Set this to **y** to send the service list, stored in the client computer's ASERVICE object, to the 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 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).
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`), does not display the user logon panel if using 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=ACROBAT
```

Install a single application with the service name of ACROBAT, while only updating the catalog. In this example, the 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=RPS,preload=Y
```

Silently preload a 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 Staging Server. 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 Configuration Server.

```
radskman
ip=10.10.10.15,port=3464,uid=RPS,preload=/opt/rps,ind=Y
```

Preload a Proxy Server using a location of `/opt/rps` 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.

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 (TIMER)

The Radia Scheduler service, **radsched**, is installed with the Application Manager. The Radia Scheduler allows you to deploy a service at a specific time. It wakes up once a minute to see if there are any scheduled items to execute.

The radsched daemon may be started as a service on UNIX workstations. We recommend running the radsched daemon as root. Please consult your company's Radia system administrator for more information.

The information about when to deploy the service is stored in two places. First, the time and date are configured in the Scheduler (TIMER) instance in the SOFTWARE domain in the Radia Database on the Configuration Server. The next time the client computer connects to the Configuration Server, the ZTIMEQ object is created on the client computer, and the timer variables are transferred to the ZTIMEQ object.

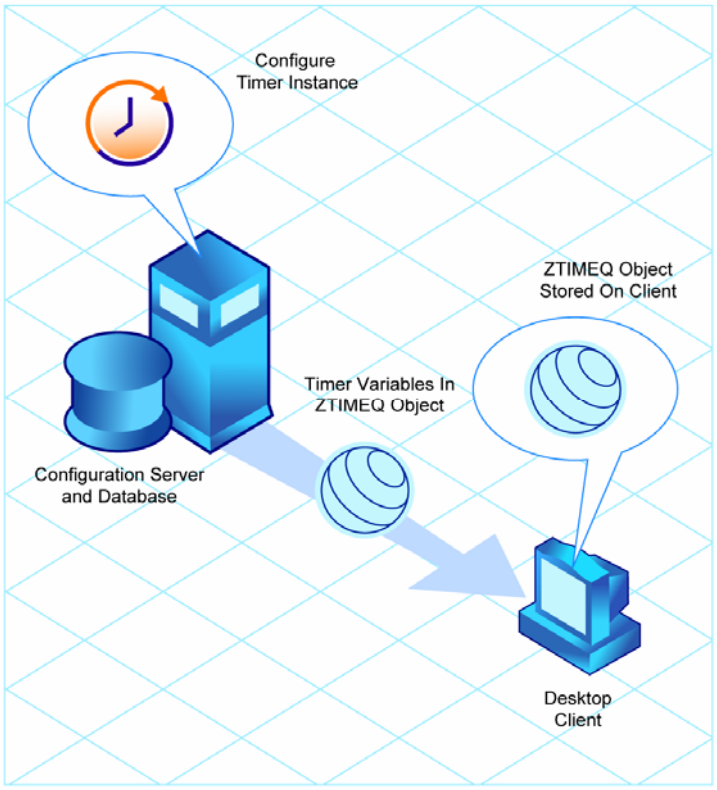


Figure 16: Transferring the timer instance.

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 Application Manager client is the scheduled deployment strategy. This strategy installs an initial set of mandatory applications when you install the 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 System Explorer to create a Scheduling (TIMER) instance in the SOFTWARE domain.

To create a new timer in the SOFTWARE domain

1 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

The System Explorer Security Information dialog box opens.

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.

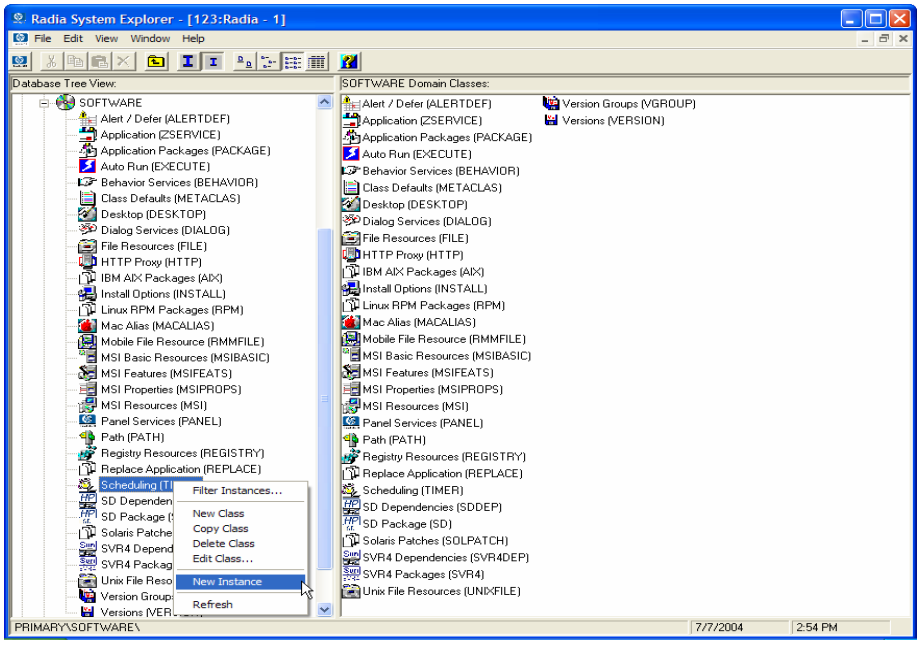
2 If necessary, type a User ID and Password, and then click **OK**.

3 The System Explorer window opens.

4 Double-click **PRIMARY**.

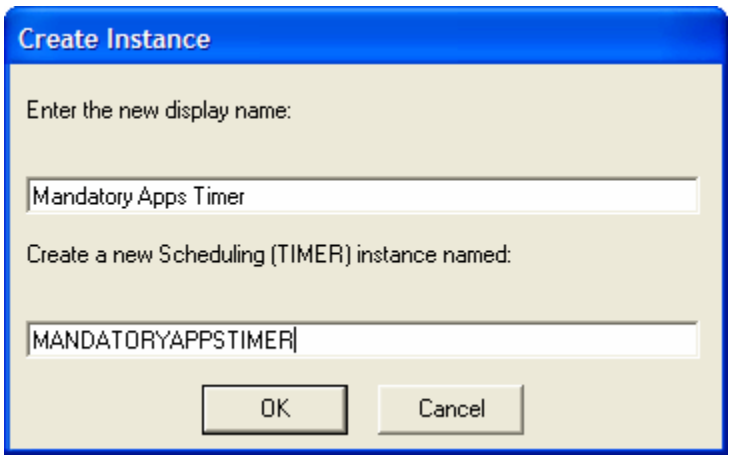
5 Double-click **SOFTWARE**.

6 Right-click **Scheduling (TIMER)**.



7 Select **New Instance**.

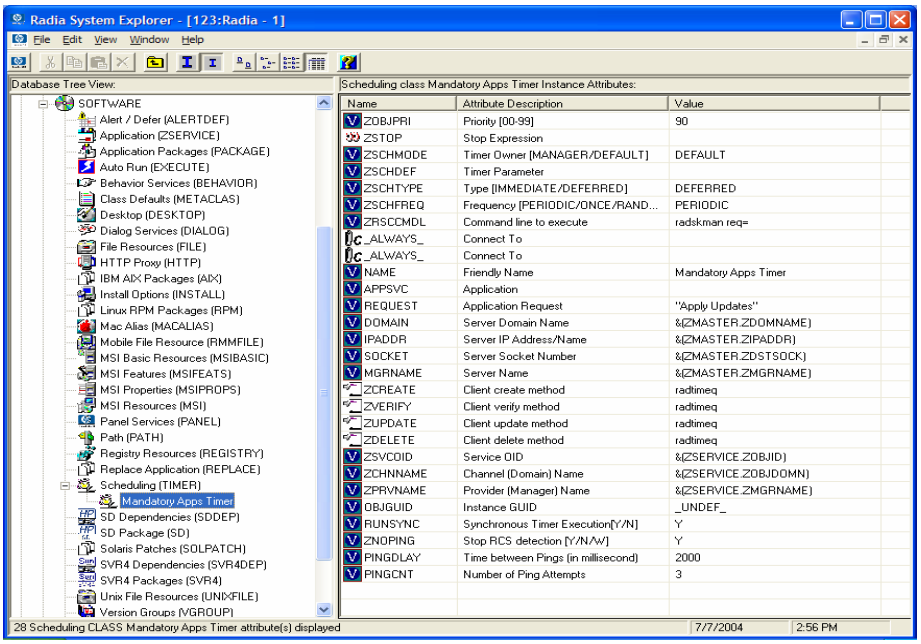
8 The Create Instance dialog box opens.



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.



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 34: Scheduling (TIMER) Attributes

Attribute	Usage
ALWAYS	Stores connections to other instances.
NAME	The friendly name for this instance.

Attribute	Usage
NETAVAIL	(Windows Only) 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 three 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 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

Attribute	Usage
ZNOPING	Use this attribute to control automatic sensing of a network connection between the client computer and the Configuration Server. The default is Y. An expired timer continually evaluates whether communications with the 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 Configuration Server. • Set ZNOPING to Y if you want to prevent the Scheduler service from pinging the Configuration Server. This is especially useful for mobile users. • Set ZNOPING to N if you want the Scheduler service to ping the Configuration Server. If the 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 Configuration Server again. If the 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 Configuration Server again. • Set ZNOPING to W if you are specifying an end limit in the ZCHDEF attribute. The Scheduler will ping the Configuration Server before executing the command. If the 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.

Attribute	Usage
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) on page 255 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 278 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 281 for more information.
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 4 A.M. (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.

Attribute	Usage
	• 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.	
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.

Attribute	Usage
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.
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 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 35: Syntax of ZSCHDEF Attribute

* Name of Weekday is the name of a specific weekday, e.g., Monday.

Attribute	Usage	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.
HOURLY	HOURLY(&ZSYSDATE, 08:41:00)	Hourly starting at 8:41 AM on system's date.
INTERVAL	INTERVAL(&ZSYSDATE, 08:41:00,,30)	Every 30 minutes starting at 8:41 A.M. 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.

Attribute	Usage	Timer Expires
MONTHDAY	MONTHDAY(*Name of Weekday**,01:00:00,,2)	Every *Name of Weekday** at 1:00 A.M. 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

Attribute	Usage	Timer Expires
		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 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.

If the ZSCHDEF is WEEKDAY(TUESDAY,*start*,*end*) then the Scheduler is going to execute it on TUESDAY, and only on TUESDAY. If *start* is 22:00:00 and *end* is 05:00:00, then the *end* time is no longer on TUESDAY, and therefore not a valid time. The RADTIMEQ create method enforces this by not allowing a RANDOM time to be created for WEEKDAY (or MONTHDAY) if the time spans midnight.

If the ZSCHDEF is WEEKDAY(TUESDAY,*start*,*end*) then the Scheduler is going to execute it on TUESDAY, and only on TUESDAY. If *start* is 22:00:00 and *end* is 05:00:00, then the *end* time is no longer on TUESDAY, and therefore not a valid time. The RADTIMEQ create method enforces this by not allowing a RANDOM time to be created for WEEKDAY (or MONTHDAY) if the time spans midnight.

To do this, the syntax for ZSCHDEF is:

```
ZSCHDEF =<frequency>(<date>, <from_time>, <to_time>, <limit>)
```

Table 36: 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 P.M. and 7:00 P.M.

To specify when the timer expires

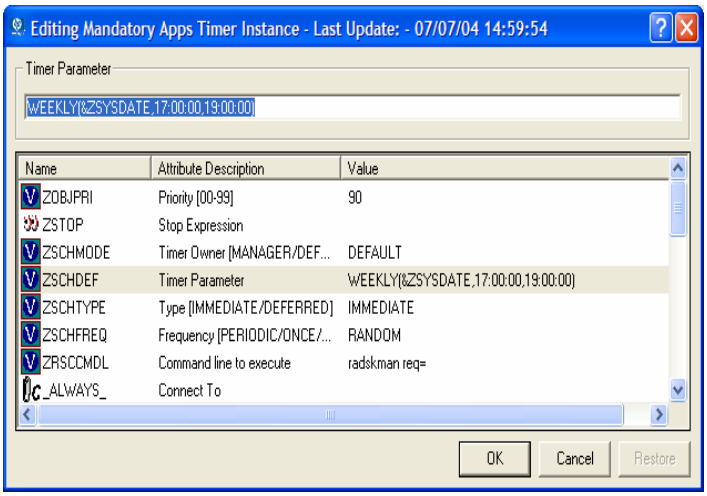
1 After navigating to the timer instance, double-click **ZSCHFREQ** in the list view of the System Explorer.

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)`**.



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.

OR

8 Select the next attribute to edit.

Specifying the Command Line (ZRSCCMDL)

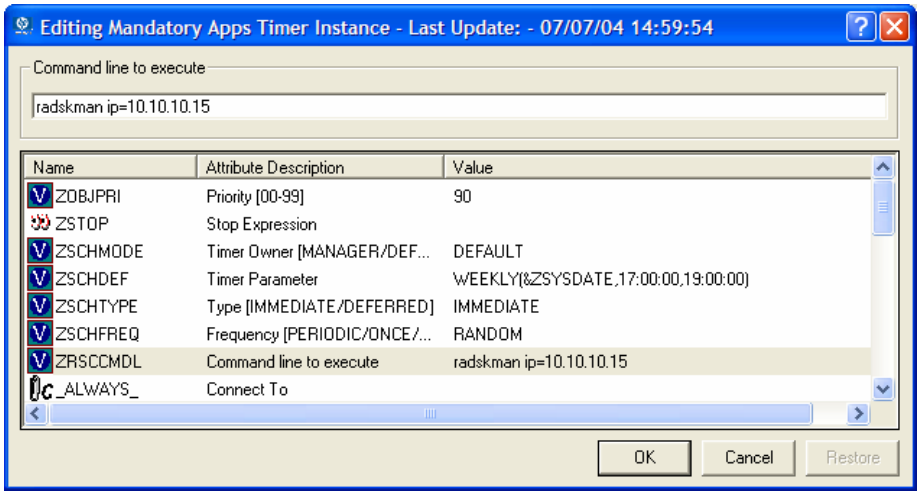
When the timer expires, it executes any command line that you specify on the client computer.

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 System Explorer.

The Editing Instance dialog box opens.



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 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,

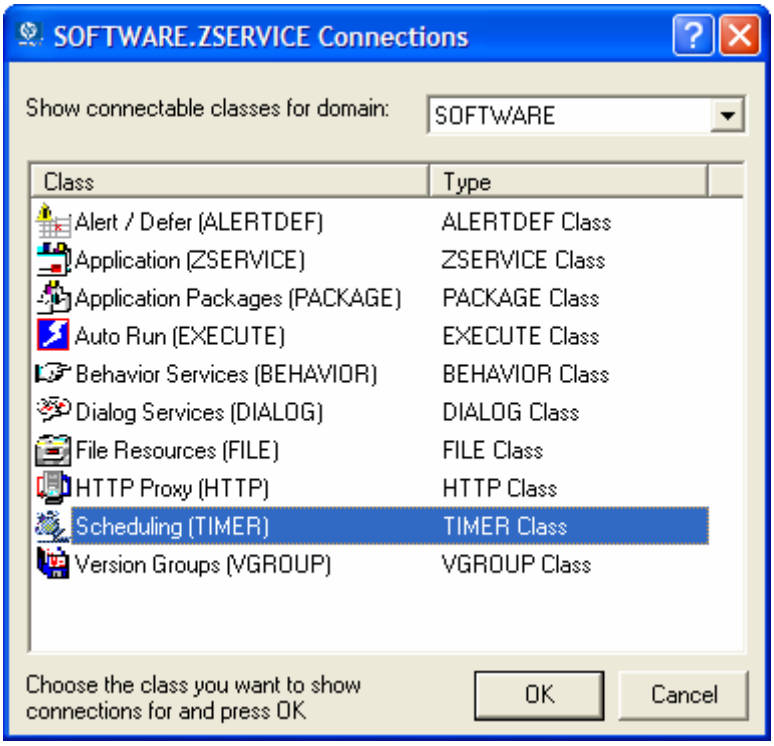
2 In the System Explorer, double-click **PRIMARY**.

3 Double-click **SOFTWARE**.

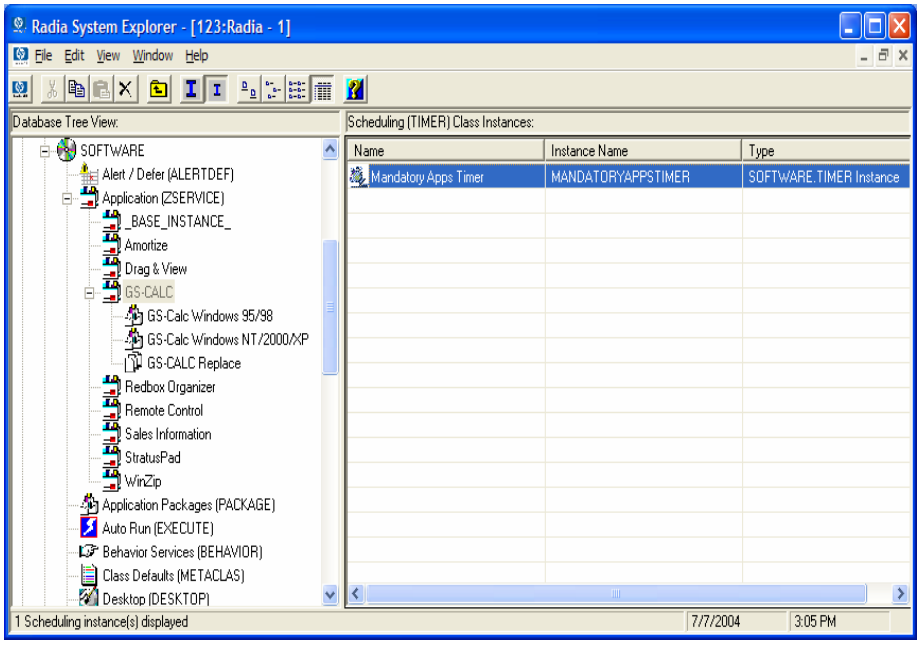
4 Double-click the **Application (ZSERVICE)** class.

5 Right-click **GS-CALC**. A menu opens.

6 Select **Show Connections**. The SOFTWARE.ZSERVICE Connections dialog box opens.



7 Select **Scheduling (TIMER)**, and then click **OK**. The Timer class instances appear 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 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, `/opt/Novadigm/lib/`).

In this section, we will force the client computer to connect to the 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 Configuration Server

1 On the client computer, go to a command prompt and change the directory to `/opt/Novadigm`. This is the default location for radskman.

2 Type **`radskman ip=<manager ip>,port=<mgr_port>`**

See Connection Parameters (radskman) on page 255 for information about radskman and the parameters above.

If you plan to do further testing, consider creating a batch file that contains the command line. Save the file in IDMSYS on the client computer. Then, create a shortcut on the desktop of the client computer.

3 Press **Enter**. 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 Configuration Server, the ZTIMEQ object is stored on client computer. Use the Client Explorer, installed as part of the Administrator Workstation, to view or modify the ZTIMEQ object.

After the timer expires, the ZTIMEQ object is removed from the client computer during the next Client Connect.

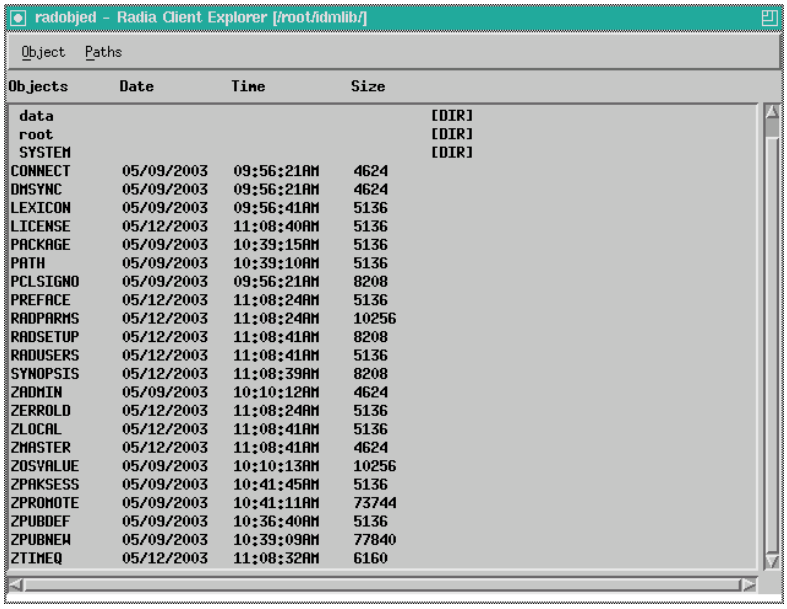
If the timer is configured to expire only one time, in the TIMER.ZSCHFREQ attribute, it will be removed immediately after the timer expires, during the next Client Connect. If the timer is configured to expire more than one time, the ZTIMEQ object will be removed after the timer expires for the last 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.

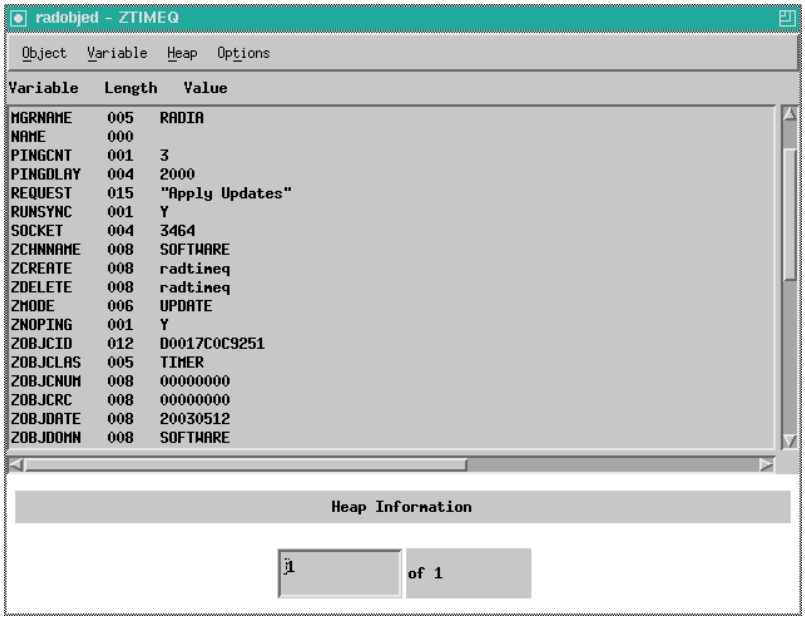
To view the ZTIMEQ object on the client computer

1 Go to the directory where you installed the Administrator Workstation.

2 Type **`./radobjed`** and press **Enter**.



4 Double-click the **ZTIMEQ** object. The ZTIMEQ object opens.



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.

The following table describes the timer logs.

Table 37: 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 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 Configuration Server. When a message is received, the client computer connects to the Configuration Server and performs the action initiated by the Notify operation. Radia Notify can also send e-mail notification to client computers.

If you are using the Management Portal, you can use the Notify Task. Refer to the *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.

Drag-and-Drop Notify is intended for use in environments with a single Configuration Server. If you are working in an environment with multiple Configuration Servers, consider using the Management Portal. Contact your sales representative for details.

Requirements for Using Notify

To use Notify

- The client computer must connect to the Configuration Server prior to the notification. This populates the PROFILE file, which contains the client computer's network address, used by Notify.

Notify is designed to notify only subscribers whose information is in the PROFILE file in the Radia Database.

- Confirm that the Configuration Server Settings file, EDMPROF.DAT, is configured properly, as shown in the code sample below. This file and the relevant lines are created in the 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
CMD_LINE=(zrtrymgr) RESTART=YES
```

- 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 `/opt/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 Configuration Server is properly configured for email.

Initiating a Notify from a ZSERVICE Instance

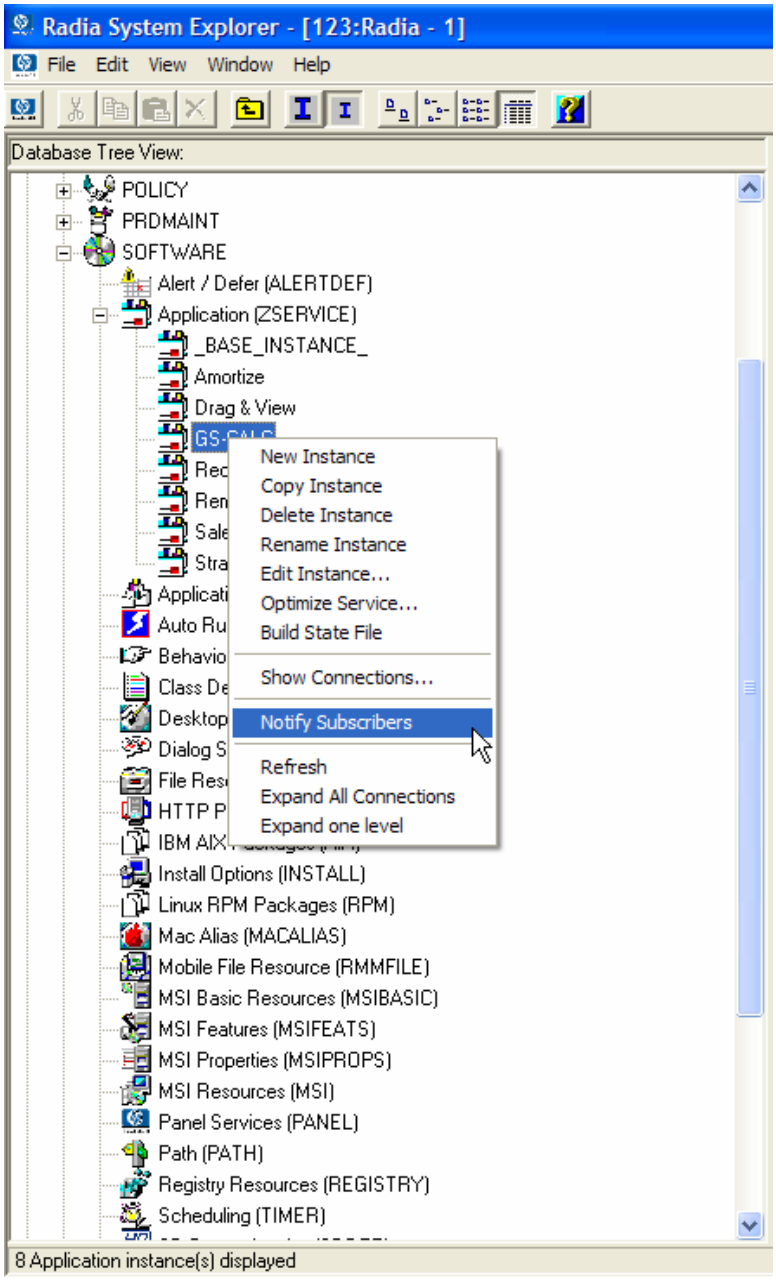
To update or remove a service, initiate the Notify from the shortcut menu for the Application (ZSERVICE) instance.

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

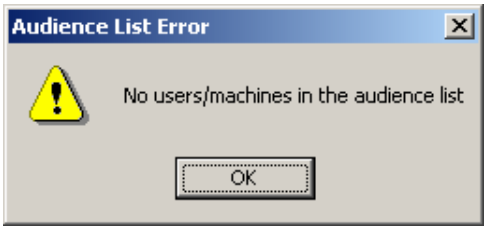
1 Right-click the **Application (ZSERVICE)** instance, such as **GS-CALC**.

A shortcut menu opens.



2 Select **Notify Subscribers**.

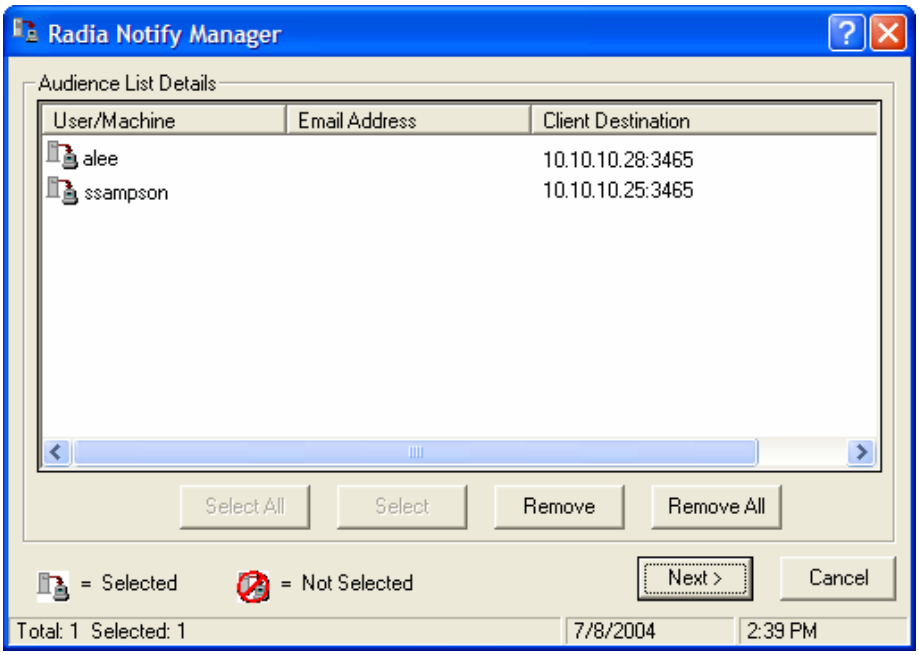
The Notify retrieves the list of the subscribers from the POLICY domain. If the selected application does not have any subscribers, the following message appears.



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.



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. Refer to the *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)

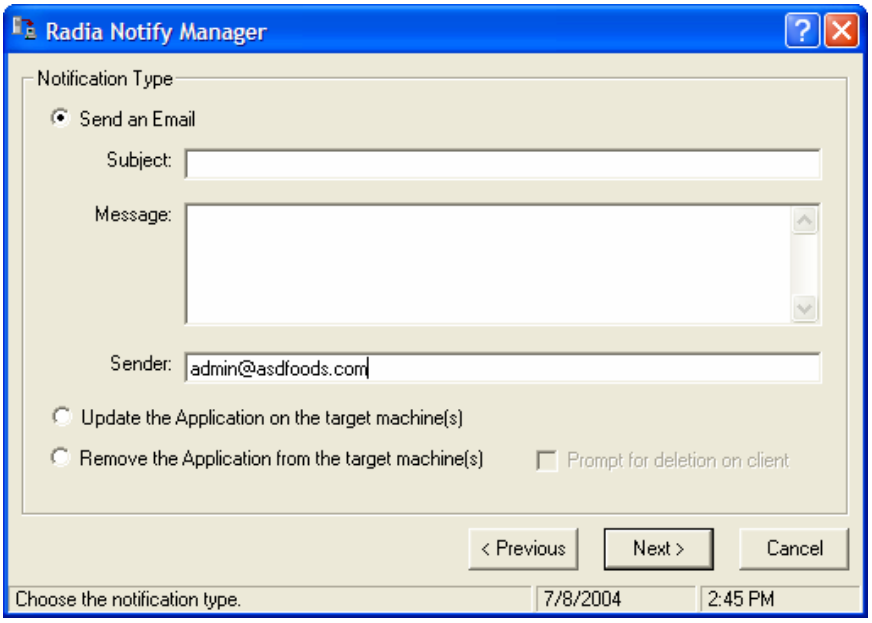
Select this option to install updates or new versions of an application on the client computers.

— 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.



6 Click **Next** to continue.

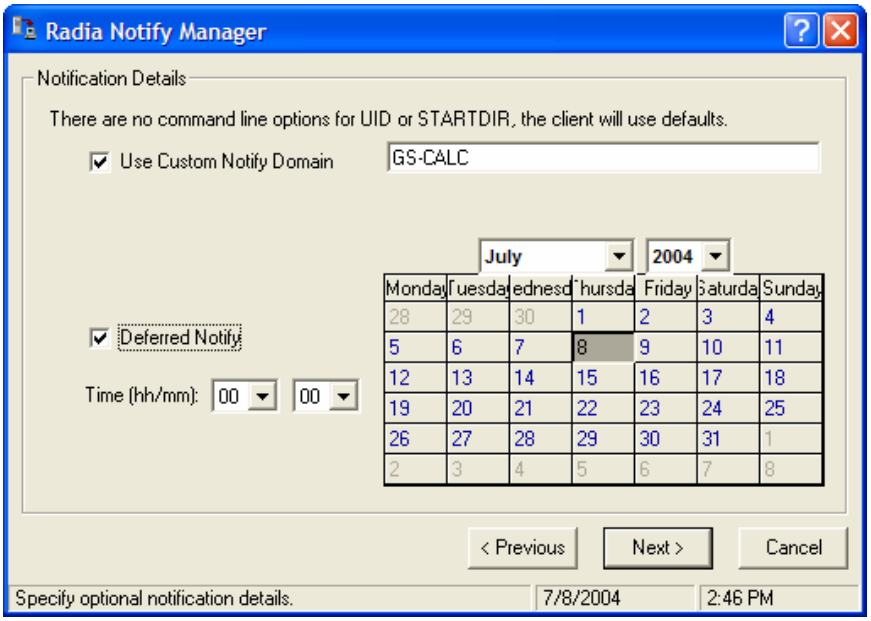
The Notification Details dialog box opens.

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.

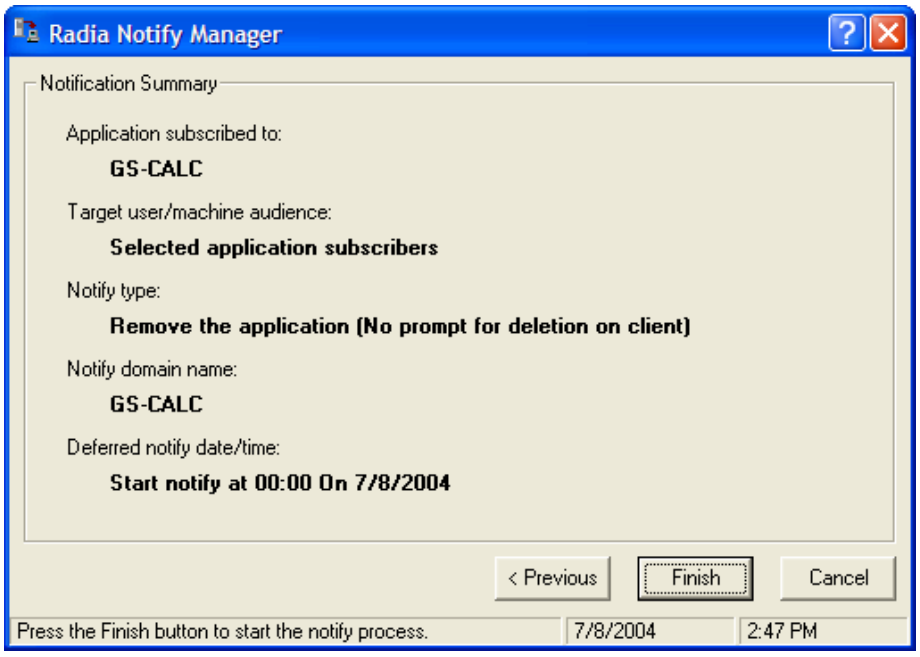
— Select **Use Custom Notify Domain** and type a name in the text box for the new domain, located in the NOTIFY file, in the Radia Database.

— Select **Deferred Notify** and use the **Time (hh/mm)** drop-down lists and the calendar controls to schedule the Notify.



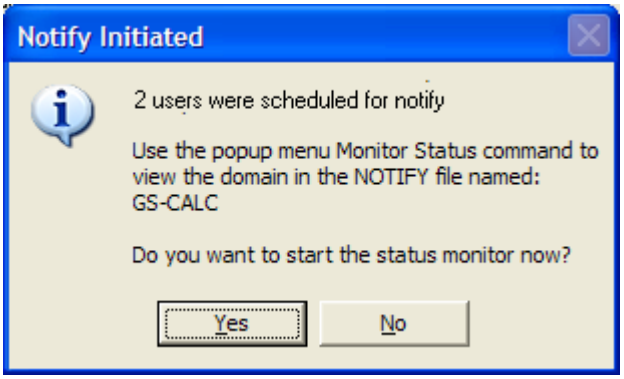
7 Click **Next**.

The Notification Summary dialog box opens.



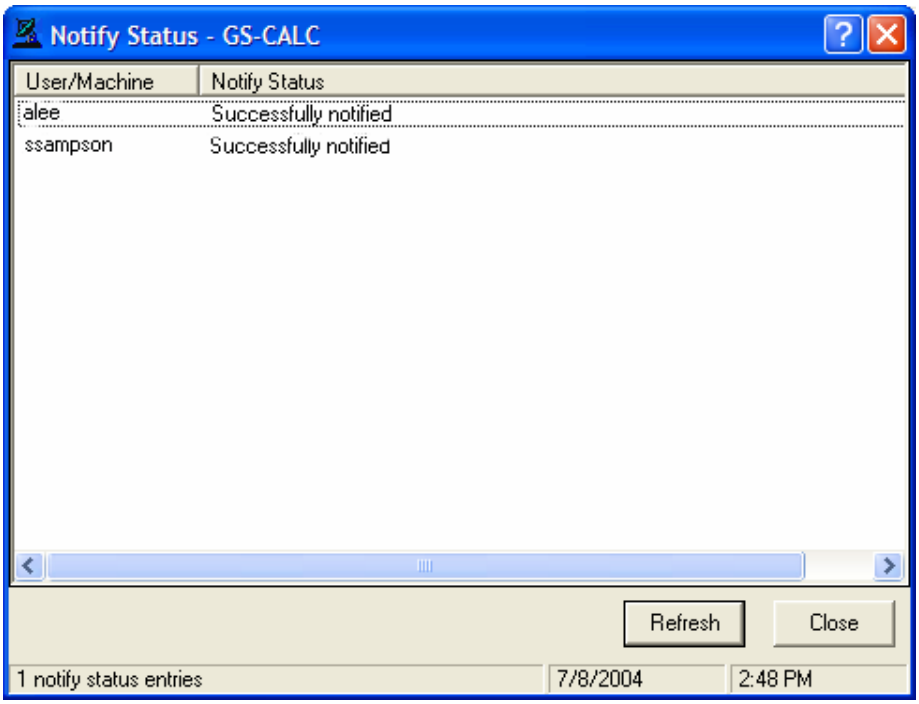
8 Click **Finish** to begin the Notify.

A message asks if you want to start the status monitor.



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.



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 Configuration Server prior to the notification. This populates the PROFILE file, which contains the client computer's network address, used by Notify.

The Drag-and-Drop Notify is intended for use in environments with a single Configuration Server. If you are working in an environment with multiple 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 Go to **Start** → **Programs** → **Administrator Workstation** → **System Explorer**.

The System Explorer Security Information dialog box opens.

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 System Explorer window opens.

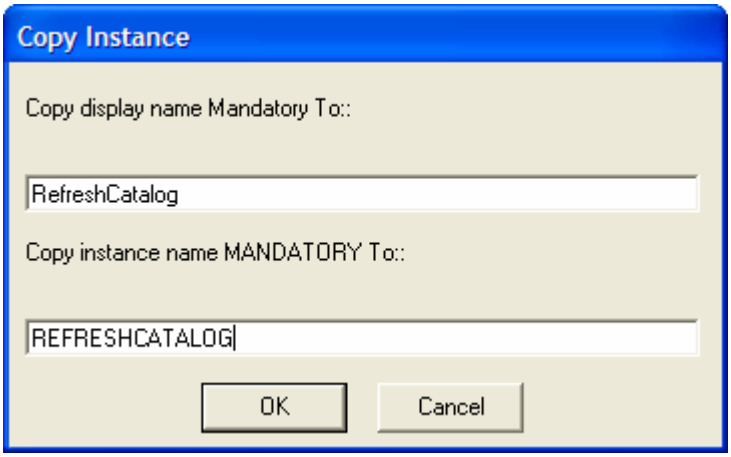
3 Double-click **PRIMARY**.

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.



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**.

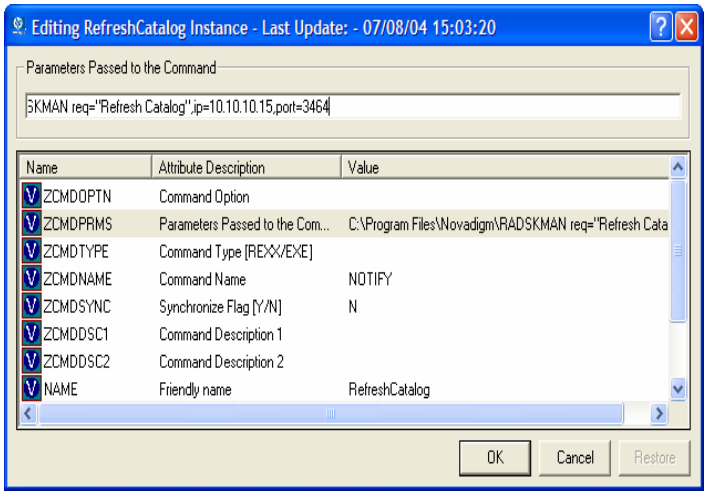
The **RefreshCatalog** instance appears in the list of ZCOMMAND class instances.

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.



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 283 for more information.

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 `/opt/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.

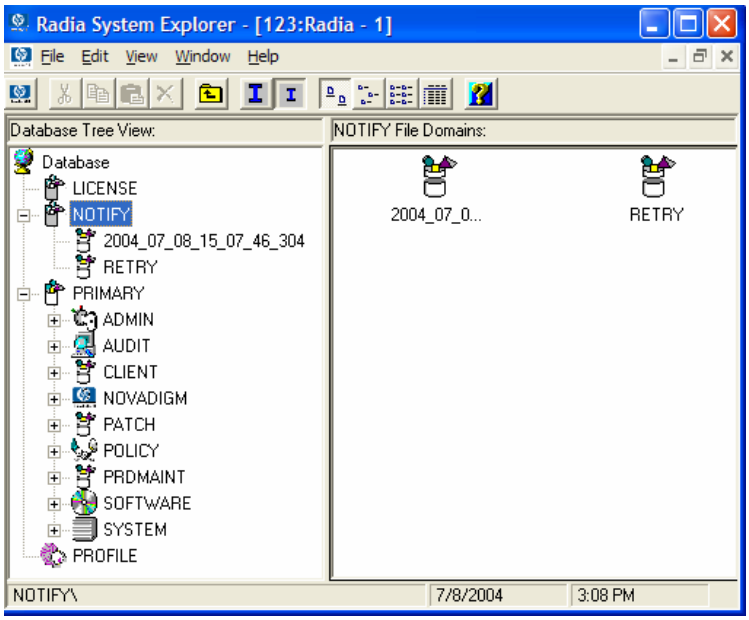
The Notify is sent immediately to the specified subscribers and the command line in ZCMDPRMS is executed.

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.



By default, Radia automatically retries the Notify operation for failed attempts. To do this, the Configuration Server is started with the Notify Retry Manager (zrtrymgr module), as indicated in the following excerpt from the Configuration Server Settings file, `EDMPROF.DAT`, located in the bin directory of your Configuration Server's installation directory.

```
[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
```

If you make any changes to the Configuration Server Settings file, `EDMPROF.DAT`, you must restart the Configuration Server service.

For more information on editing the Configuration Server Settings file, see the *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.

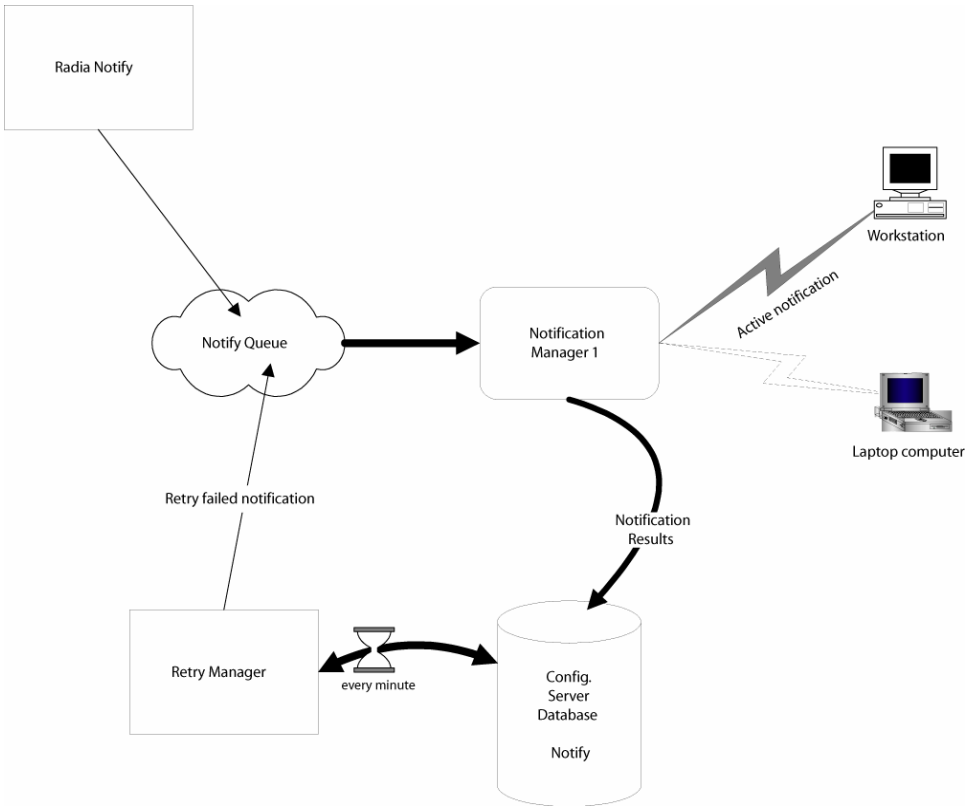


Figure 17: Retry process.

Viewing the Results of a Notify

You can find information about a Notify:

- In the System Explorer in the NOTIFY file.
- In the Status Monitor accessed from the NOTIFY file in the System Explorer.

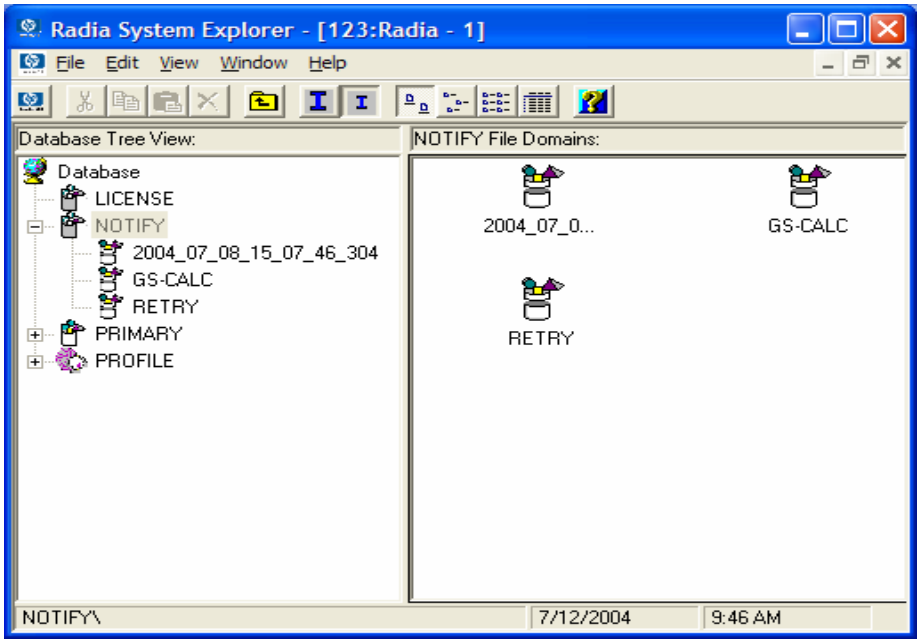
You can also find information about a Notify operation in the Configuration Server log. The log file is stored on the Configuration Server in the LOG directory.

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 System Explorer, double-click **NOTIFY**.



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 38: 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.

Attribute	Description
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.
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

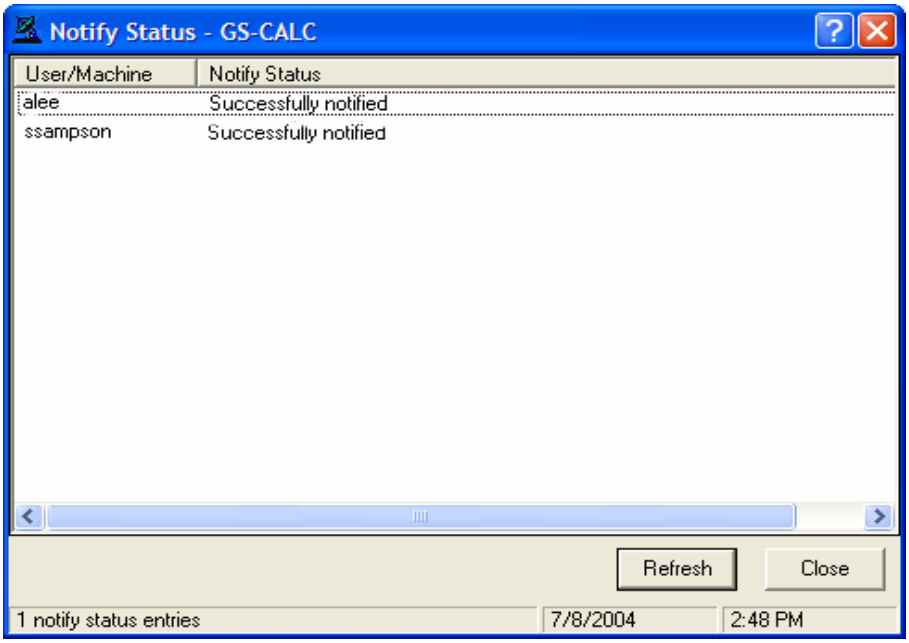
Use the Status Monitor to review the results of a Notify or Retry operation for all of your subscribers.

To see the status of a Notify or Retry

1. In the 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.



4 Click **Refresh** if you think the status might have changed.

OR

Click **Close** to close the Status Monitor.

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.

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 *Application Manager* or *Software Manager Guide* for your version of the Radia client.

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 (P). Maintenance *only* runs when the mnt parameter of radskman is set to Y. See *ACTMAINT* on page 206 and the *Mnt* parameter on page 266 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 Configuration Server service and copy the export files to the Configuration Server's `bin` directory.
4. Import the files using the ZEDMAMS utility. For detailed information on the use of this utility, refer to the *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
```

Your command line may vary depending on a number of factors. For detailed information on the use of this utility, see the *Configuration Server Guide*.

5 Restart the Configuration Server.

6 Assign the Maintenance Service to the appropriate users in the POLICY domain.

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.

About Proxy Servers

Use Proxy Servers to load a portion of the work required to deploy applications from the Configuration Server to another server computer. You may want to do this for the following reasons:

- The Proxy Server may be closer to the clients on the network.
- You may want to reduce the load on the Configuration Server.

When using Proxy Servers, the software to be distributed is copied to the Proxy Server. The Proxy Server then provides the software to those Radia clients that are not required to obtain their software from the Configuration Server. The potential benefit of Proxy Servers must be evaluated individually for each server and its Radia subscriber computers.

For more information, refer to the *Proxy Server Guide*.

Summary

- Carefully plan and test your application deployment strategy to determine the best distribution method for your subscribers.
- Use the Scheduler service to deploy an application at a specific time or interval.
- Use the Notify function to update or remove an application that has already been deployed, or to notify users via e-mail of an update.
- Consider if you have any special cases for deployment that may need further configuration.
- Use the Version Group Editor when you have multiple versions of the same application. You can use the Version Group Editor to schedule deployments, and set versions to activate.
- Consider using Staging Servers to minimize network traffic or to minimize work on the Configuration Server.

10 Radia Client Objects and Directories

At the end of this chapter, you will:

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.
- Know how Open Database Connectivity (ODBC) can help you generate reports with information from the objects.

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 Software Manager. This chapter covers Radia client objects.

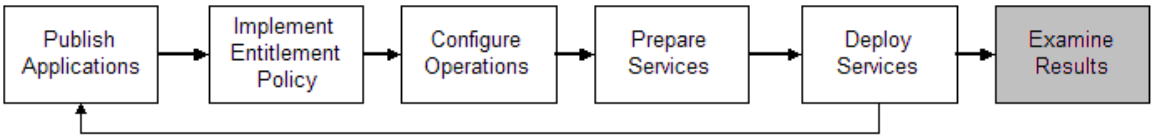


Figure 18: Tasks completed in this guide.

Radia Client Directory Structure

Below is an example of the directory structure on a client computer following a forced system connection from the client to the Configuration Server, and the installation of a mandatory application. (Directories in the Table below are preceded with `/opt/Novadigm/` by default).

Table 39: Client directories

Directory (/opt/Novadigm/..)	Description
/opt/Novadigm	Client Directory (IDMSYS)
../CACertificates	SSL Certificates
../lib	LIB Directory (IDMROOT)
../lib/MAINT	Maintenance Storage Directory
../lib/BACKUP	Upgrade Maintenance Backup folder.
../lib/data	Data Storage (IDMDATA)
../lib/SYSTEM	Starting Directory (startdir) created during connect. (Name will vary).
../lib/SYSTEM/RADIA	Configuration Server name (mname).
../lib/SYSTEM/RADIA/SOFTWARE	Directory Name (dname)
../lib/SYSTEM/RADIA/SOFTWARE /ZSERVICE	ZSERVICE Class

Directory (/opt/Novadigm/..)	Description
../lib/SYSTEM/RADIA/SOFTWARE /ZSERVICE/DRAGVIEW	Sample application directory
../log	Log directory (IDMLOG)

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

Application Manager Directories

The initialization settings for the Application Manager for Windows are located in the `.nvdrc` file on the client computer. This is located, by default, in the home directory of the account used to install the client.

Table 40: NOVAEDM Parameters

Parameter	Description
IDMDATA	When Radia installs software, the Radia client temporarily stores compressed files received from the Configuration Server in this folder. Once the files are decompressed and installed on the client computer, the compressed files are erased. Default: `/opt/Novadigm/lib/Data/`
IDMLIB	Dynamic directory that stores the objects for the service currently being managed. Default: `/opt/Novadigm/lib/`
IDMSYS	Stores the Radia client executables, such as `.EXE` and `.DLL` files. Default: `/opt/Novadigm/`
IDMROOT	The base directory for IDMLIB. This is a static path. Default: `/opt/Novadigm/lib/`
IDMLOG	Stores the Radia client logs. Default: `/opt/Novadigm/log/`

About Radia Client Objects

When a client computer connects to the Configuration Server, information is exchanged between the client and the Configuration Server. This exchange is called **resolution**. During resolution, Radia checks the status of services, and updates the 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 Software Manager client and connecting to the 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 41 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 41: Core Client Objects

Object	When created or updated	Type of Information included
ZCONFIG	ZCONFIG is created at start of Client Connect process. See Table 42 on page 318 for more information.	Contains basic hardware information for the client computer such as processor, operating system, and drives.

Object	When created or updated	Type of Information included
SYNOPSIS	This object is transferred to the Configuration Server at the end of the Client Connect. Note: Client Operations Profiles must be enabled for this object to be present. See Table 43 on page 319 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 44 on page 321 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.

Object	When created or updated	Type of Information included
PREFACE	PREFACE is sent to the 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 45 on page 323 for more information.	Contains core information about each invocation of radskman.
SMINFO	(Linux Only) SMINFO is created during Client Operations Profiles resolution, but does not require Client Operations Profiles. See Table 46 on page 324 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 Administrator Workstation. Use it to view objects in the IDMLIB directory. You can view

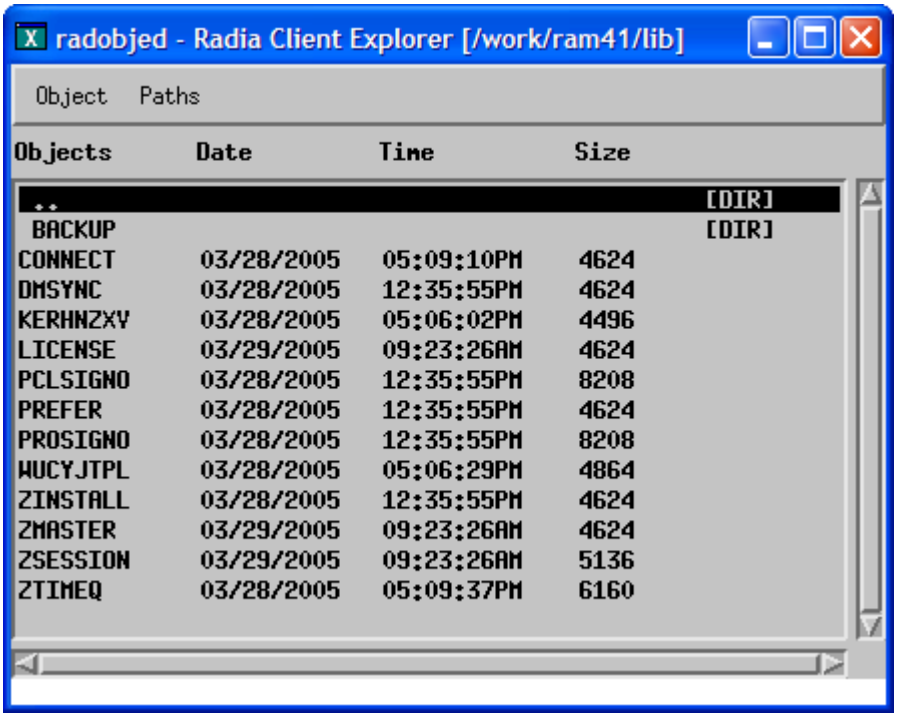
any object if you have access to the client computer's IDMLIB directory. Otherwise, you may need to manually retrieve the object file, and store it on your Radia administrator computer.

To view an object using the Radia Client Explorer

1 Go to the directory where you installed the Administrator Workstation.

2 Type `./radobjed` and press **Enter**.

The Radia Client Explorer opens.



3 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.

4 Double-click the object's name in the list view.

The Radia Client Explorer displays the selected object.

5 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.

The ZCONFIG object is sent to the Configuration Server automatically for viewing with the System Explorer. If you do not want this object sent to the 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 42: ZCONFIG Attributes

Attribute	Description	Example
RUNLEVEL	Current run level at time of scan	5
ZHDWCOMP	Computer Name	qalinux
ZHDWCPU	CPU type	I686
ZHDWD00	Drive name for drive 00	/dev/hda2
ZHDWD00F	Current free space on drive 00	26913026048
ZHDWD00M	Mount Point for Drive 00	/
ZHDWD00T	Total space for drive 00	35152932864
ZHDWDNUM	Number of drive letters assigned	3
ZHDWMACH	Machine Type	i686
ZHDWMEM	Total physical memory (RAM)	133,619,712
ZHDWOS	Operating system	Linux
ZHDWSVCP	Service pack applied	2.4.20-8
ZHDWXHID	Host ID (output of hosted comman)	771039E4
ZHDWXHN	Host Name	qalinux
ZOBJNAME	Name of Object	HARDWARE_SCAN

Attribute	Description	Example
ZOSMAJOR	Major Component of OS version	2
ZOSMINOR	Minor Component of OS version	4
ZOSREV	OS revision (output of uname –v)	#1 Thu Mar 13 17:54:28 EST 2003
ZOSVER	OS version (output of uname –r)	2.4.20-8
ZUSERID	User ID or computer name	LINUXUSER

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. Refer to the Configuring Client Operations Profile section for more information.

Table 43: 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 the *Management Applications Messages and Codes Guide*.
PRIORAPP	Total number of applications that existed in the service list (installed/not installed) before this job started
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

Attribute	Description
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

Attribute	Description
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 Configuration Server Database along with the following usage related variables. For more information on the SAP class, see the Configuring Client Operations Profile section for more information.

Table 44: 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.

Attribute	Description
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
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 - Configuration Server DATA - Proxy Servers, 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 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 45: 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
CTYPE	Type of Radia client. The possible values are: RSM Software Manager RAM Application Manager RPS Proxy Server or 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.

Attribute	Description
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 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)

Systems Management information is currently available for Linux only.

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 46: SMINFO Attributes

Attribute	Description
ASSETTAG	Unique Asset Tag number of the client computer from the BIOS

Attribute	Description
COMPDOMN	Computer Domain
COMPNAME	Computer Name
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

Controlling Default Permissions for Directories and Objects

By default, directories created by CM are assigned permissions 0777 (read, write, execute for everybody) and objects are assigned 0666 (read, write, for everybody). In order to change the default permissions assigned when new directories and objects are created by CM within IDMLIB, you can use environment variables or you can create a DEFAULTS.EDM file in IDMROOT.

Note that environment variables will always take precedence. If the environment variables are set and a `DEFAULTS.EDM` file exists, values defined using the environment variables are used.

These methods for controlling permissions apply only to newly created, service-related directories and objects within IDMLIB. For example, `/opt/OVCM/lib/SYSTEM/RADIA/SOFTWARE/ZSERVICE/SAMP_APP/00000000.000`.

To control permissions using environment variables

- Set the following environment variables with the permissions you want assigned by default:

— For directories: IDMLIBPERM

— For objects: IDMOBJPERM

To control permissions using DEFAULTS.EDM

1 Within IDMROOT, create an object, `DEFAULTS.EDM`.

2 Add the following variables with the permission value to be used when new objects or directories are created by CM.

— For directories: LIBPERM

— For objects: OBJPERM

For example, to exclude write permissions for objects for group and other, create a `DEFAULTS.EDM` file with the following:

OBJPERM 0755

To set the default permissions of directories to read and write for everybody, create a DEFAULTS.EDM file with the following:

LIBPERM 0666

The PROFILE File

Some client objects such as ZCONFIG and ZMASTER are sent to the 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 47: 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.

Instance	Information Recorded
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 the System Explorer to view the PROFILE file.

Reporting with Inventory Manager

The 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 Inventory Manager consists of two parts, the Inventory Manager server and the Inventory Manager client. The Inventory Manager server enables centralized reporting and administration based upon the discovery results. The Inventory Manager client discovers configuration information on the client device, and reports the results to the Inventory Manager server. The results are sent to the 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 Inventory Manager reports from either the Radia Integration Server or from the Management Portal, if installed.

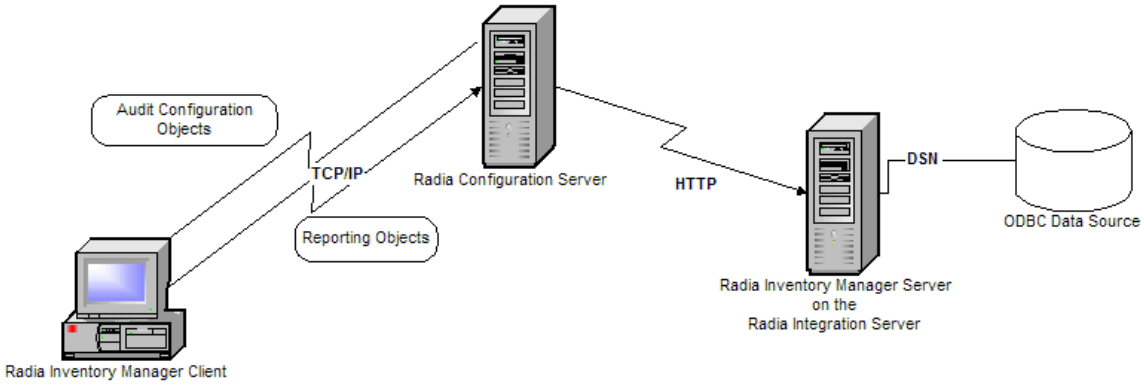


Figure 19: The Inventory Manager communicates with an ODBC source.

The Inventory Manager provides the following benefits:

- Audits hardware information.
- Audits software, including the ability to audit specific files or classes.
- Collects files from client devices.
- Generates reports on collected information through an ODBC-compliant database.
- Reports on Personal Digital Assistants (PDAs).

The Inventory Manager is one module of the Radia Integration Server. The Radia Integration Server stores and maintains the information discovered by the Inventory Manager in an ODBC-compliant database, and uses active server pages to provide reports from the Radia Database. For computers with WBEM services installed the Radia Inventory Management client can query the WBEM namespace and return information such as hardware, file system, daemons, and event logs. For computers that do not have WBEM services installed, the Inventory Manager uses Radia client objects to generate information on the file system and hardware. Refer to the *Inventory Manager Guide* for additional information.

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 it is located in

the IDMLOG location. 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 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=/opt/Novadigm/lib
```

See the HP OpenView web site for a technical note 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 Inventory Manager.

A 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 System Explorer groups instances based on the identifier that precedes the underscore.

If you decide to use a high-level identifier *without* an underscore (_), you can use the System Explorer's filtering capabilities to display only the instances with that identifier.

See the *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 the next figure.

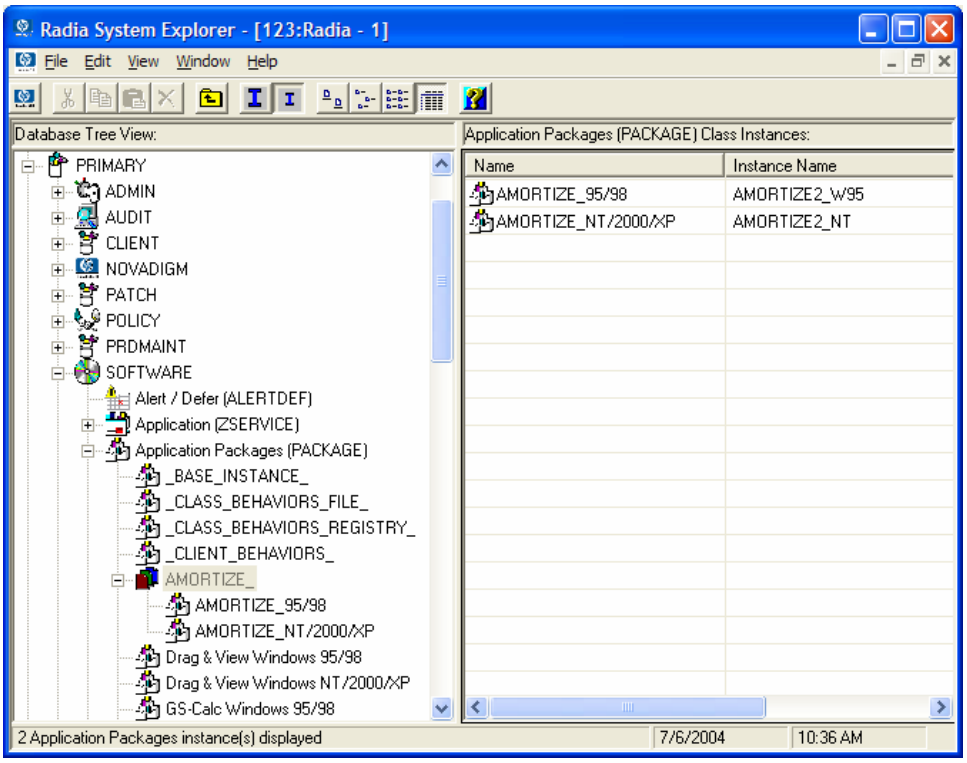


Figure 20: Instances grouped by identifier.

Naming Conventions for the POLICY Domain

We recommend that you use a variation of the following standards.

Table 48: 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 49: Naming Conventions for the USER 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 50: 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 51: 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.

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 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 Packager 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 Configuration Server and client computer. Each attribute defined in a class template has a set of 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 (Application Manager or Software Manager) runs on the client computer. It communicates with the 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 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 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 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 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 Inventory Manager.

Manager

See 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 Configuration Server methods run on the Configuration Server computer.method variable

The method variable identifies the method, or program, to be executed as part of the resolution process.

For 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. Configuration Server methods are located in the 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 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 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 Packager, 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.

Application Manager

The Application Manager (radia_am.exe) is the Radia client that manages mandatory services. The Radia administrator uses the System Explorer to specify the services that the Application Manager manages on the client computer. No user interface is available.

Radia Client Explorer

The Radia Client Explorer, installed with the 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).

Configuration Server

Also called Active Component Server or Manager.

The Configuration Server distributes applications to client computers. It runs on the server and maintains the Radia Database, which stores information that the 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 System Explorer to manipulate the Radia Database.

Inventory Manager

The 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 Packager

The Radia Packager is used to create packages of data and store them in (i.e., promote them to) the Radia Database.

Proxy Server

When used in your environment, the Proxy Server can reduce the load on your Configuration Servers by distributing requested resources to client computers. The 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 Application Manager, allows you to deploy a service at a specific time.

Radia Screen Painter

The Radia Screen Painter, installed with the Administrator Workstation, is used to develop custom dialog boxes.

Software Manager

The Software Manager (radia_sm.exe) is the Radia Client used to manage optional services. The Radia administrator uses the System Explorer to specify the services that are available to the subscriber.

The subscriber installs and manages data that is available from the Software Manager user interface.

Staging Server

The Staging Server is used to store data required for deploying applications on a computer other than the computer with the Configuration Server.

System Explorer

The System Explorer, installed with the Administrator Workstation, is used to manipulate the contents of the Radia Database.

resolution

Resolution occurs when the 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 Configuration Server. The 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 Packager that results in the creation of one Radia package.

Software Manager

See Software Manager.

Staging Server

See 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 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.

Index

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z