HP Client Automation

Application Manager and Application Self-service Manager

for Linux and Mac OS X operating systems

Software Version: 7.80

Installation and Configuration Guide

Manufacturing Part Number: None

Document Release Date: April 2010

Software Release Date: November 2009

Legal Notices

Warranty

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

The information contained herein is subject to change without notice.

Restricted Rights Legend

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

Copyright Notices

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.

DHTML Calendar
Copyright Mihai Bazon, 2002, 2003

opyright Mihai Bazon, 2002, 2003

Documentation Updates

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

- Software Version number, which indicates the software version.
  - The number before the period identifies the major release number.
  - The first number after the period identifies the minor release number.
  - The second number after the period represents the minor-minor release number.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

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

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

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

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

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

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

Table 1 contains changes made to this document for this release.

Table 1 Changes added

Chapter	Version	Changes
All	7.20	Configuration Management was renamed HP Client Automation. Note that not all features and components are re-branded.
All	7.20	Support for UNIX operating systems changed for this release. Refer to the Product Support information in the Release Notes for the latest operating system support information.
Chapter 2	7.20	Page 42, Installing the HPCA Agent to an HP Thin Client: new section describes how to istall the Agent to thin client devices.
Chapter 2	7.50	Page 42, Installing the HPCA Agent to an HP Thin Client: Updated instructions for installing the agent to HP thin client.
Chapter 2	7.50	Page 44, HPCA Registration and Agent Loading Facility: added new section to describe RALF.
Chapter 5	7.50	Page 115, Notify Security (NTFYSEC) is a new section that

Chapter	Version	Changes
		details the notify-security feature of HPCA.
Chapter 8	7.50	Page 209, To access the user interface for Macintosh: is a new section that describes how to lauch the Application Self-service Manager user interface for Macintosh.

Support

You can visit the HP Software support web site at:

www.hp.com/go/hpsoftwaresupport

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

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

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

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

To find more information about access levels, go to:

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

To register for an HP Passport ID, go to:

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

Contents

1 Introduction

At the end of this chapter, you will:

- Understand the components of HP Client Automation (HPCA) products.
- Be familiar with the structure of the HPCA Configuration Server Database.
- Understand suggested deployment strategies.

About this Guide

This guide describes the *suggested* implementation for the HP Client Automation Agents, Application Manager and Application Self-service 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 these HPCA Agents.

Before you can manage software, you must install the agent and the Administrator. The Administrator installation is covered in the *Administrators Guide*.

Note to Macintosh users: The Agent for Mac OS X is similar to the Agent that is installed and runs on Linux operating systems. For that reason, instructions for using the Agent for Mac OS X have been included in this guide. There are obvious differences, though. For example, installing the agent to Macintosh devices may require the sudo command as well as the default installation location for the Mac OS X Agent is, `/Applications/HP/CM/Agent`.

The *Application Manager and Application Self-service Manager Guide* describes the following:

- Installing the HPCA Agents
  This chapter describes how to install the agents.
- HPCA Agent Directories and Objects
  This chapter shows you where to find and how to examine the results of your HPCA implementation.
- 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 agents to use the most appropriate Configuration Servers and Proxy Servers, provide for fail over capabilities, and configure your agent.
- Preparing Services
  This chapter describes services options such as restarting the agent device and implementing applications that have machine- and user-specific components.

- Deploying Services
  This chapter expliains methods available for deploying services to devices in your environment.
- HPCA Application Self-service Manager User Interface
  This chapter explains how to use the Application Self-service Manager User Interface.

Using this Guide with Core and Satellite Servers

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

About Client Automation Solutions

HP Client Automation Solutions automate the management of data on target devices in your enterprise. These solutions manage devices based on parameters that you configure.

The following are some key characteristics of HPCA:

- **Desired State Approach**
  You can configure and maintain the desired state of a device's operating system, applications, and configuration. If there is a difference between the device's desired state and the current state, the HP differencing technology determines the precise component-level changes that are required and sends only those changes to the managed device.
- **Policy based Management**
  You can define entitlements, which control the deployment of data to authorized users or target devices. For example, an IT administrator may implement a policy that only permits access to certain financial databases by a select workgroup within an organization's finance department. Subsequent changes to entitlements cause data to be installed, changed, or removed for affected users or devices.
- **Adaptive Client Automation**
  As the policies change, the current state is differenced and reconfigured to correspond to the desired state.

- **End-to-End Lifecycle Management**
  You can automate the policy-based management of data throughout the deployment life-cycle. With HPCA solutions you can:
  - Package applications.
  - Analyze the impact of packages prior to deployment.
  - Discover, collect, and report on hardware and software information.
  - Configure policy assignments to assign data to the appropriate device or user.
  - Distribute and install data across enterprise networks.
  - Repair data and configurations through the desired state process.
  - Deploy patches, service packs, hot fixes, and application updates.
  - Remove data from managed devices by changing entitlement policies.

With the HPCA products, you can manage data and continuously configure devices.

Terminology

The following terms are used throughout this guide, and it may be helpful to become familiar with them.

agent

The software (such as the Application Self-service Manager, Application Manager, Inventory Manager, Patch Manager, or Application Usage Manager) that runs on a managed device and communicates with the Configuration Server.

agent connect

The process by which a managed device communicates with the HP Client Automation Configuration Server.

HPCA Configuration Server

Used in conjunction with the HPCA Configuration Server Database, a server that stores, manages, and distributes application package information, and manages policy relationships and information about managed devices. This server is the only product that is mandatory in the HPCA Solutions environment; without it, the infrastructure will not function.

desired state

The condition of a device defined by configuration parameters set in the Configuration Server Database.

device

In this document, a device is a piece of hardware, such as a computer or ATM, that may be either a managed device or a target device.

managed device

A computer, ATM, or other piece of hardware that is managed by HPCA Solutions.

package

(n) A unit of software or data that can be published to the Configuration Server Database.

(v) The process of grouping data into a unit that can be published to the Configuration Server Database.

policy

A designation of the services to which a user, a target device, or a managed device is entitled.

resolution

The process by which the object attribute values on a managed device are replaced with those that are required to achieve its desired state.

service

A group of related packages, methods, or behaviors organized into manageable units.

target device

A workstation or server on which you want to install, replace, or update software.

user

In HPCA, the identity of the device or subscriber being managed.

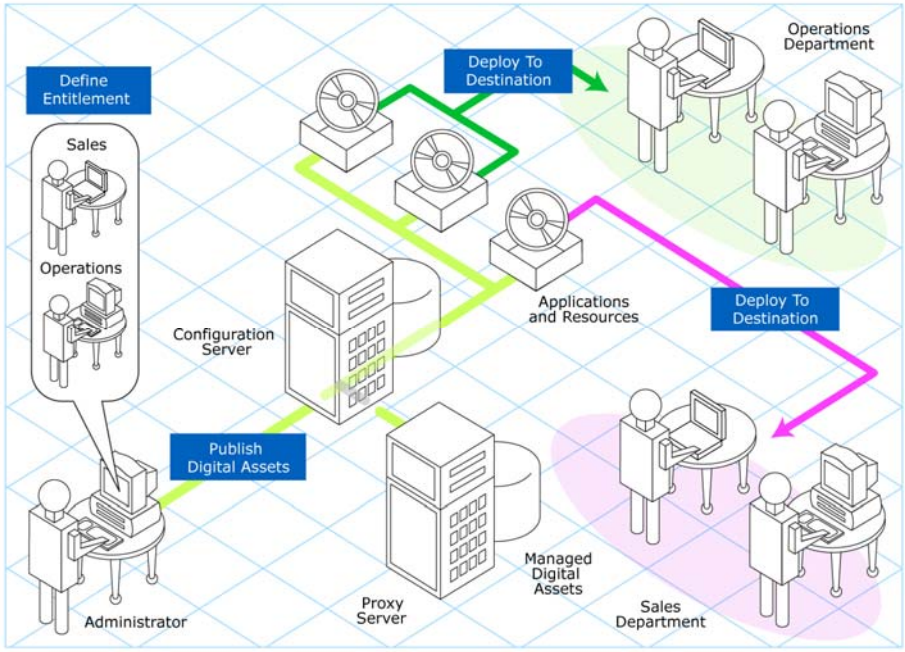
Desired State

HPCA manages the distribution of assets based on your **desired state**. This records the identities and intended configurations of your managed devices. The desired state can be simple or complex.

At a minimum, the desired state ncludes the following five elements:

- **Users**
  The identity of the devices being managed. This can be either a computer name or a user name.
- **Applications**
  The software that is being managed.
- **Application Resources**
  The components that make up the applications.
- **Deployment Source**
  The location where the application components are stored, such as on a HP Client Automation Proxy Server or on an HP Client Automation Configuration Server, so they can be distributed to the users.
- **Deployment Destinations**
  The location to which the application and its files will be distributed such as desktop computers, servers, PDAs, and laptops.

Figure 1 Elements in a desired state.



Use HPCA Products to manage all of these components. You will publish **packages** of data, determine entitlement policy, and define how the packages will be deployed.

A package is a unit of distributable software or data.

The Configuration Server Database

The HP Client Automation Configuration Server Database, located on the HP Client Automation Configuration Server, records your enterprise's desired state model. This model is composed of the data to be distributed, policies that define the services to which users or devices are entitled, and security and access rules for administrators. Refer to the *HP Client Automation Configuration Server Database Reference Guide* for information on the structure and usage of HPCA classes.

The database is hierarchically structured as follows:

- **Files** are used to group similar domains. The PRIMARY File is used to define and maintain the desired state.
- **Domains** are logical file partitions used to group similar classes. For example, the POLICY Domain contains the classes needed to create users and groups.
- **Classes** are templates containing the attributes needed to create an instance. A class represents a category of the desired state. For example, the USER Class of the POLICY Domain defines users of managed applications. It defines all of the attributes necessary to identify the managed device.
- **Instances** are actual occurrences of classes. The attributes of a class instance contain data describing one specific entity of that class. For example, a USER Instance contains the information needed to identify a target device or user.
- **Attributes** are data elements of a class. The class contains the definition (e.g., the name, data type, description, and length) for each attribute belonging to the class. Each class instance created from the class contains a value for each of the attributes defined in the class. For example, the NAME attribute of a USER Class contains the name of the user.

Default Files and Domains

When you install the Configuration Server, LICENSE and PRIMARY are the only two files available. As you use HPCA, your database may change. Some of the management infrastructure products add other domains. For example, Patch Manager adds the PATCHMGR Domain, and Application Usage Manager adds the USAGE Domain.

- The LICENSE File is read-only and is used for Configuration Server processing. This file is for HP use only, and should not be modified.
- The PRIMARY File is where you will find most information regarding software management. The PRIMARY File contains 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 assets on your devices.
  - Use the CLIENT Domain to configure Client Operations Profiles. This includes defining which Configuration Servers, and Proxy Server the managed device can use.

- Use the POLICY Domain to create users and groups, and to assign users to groups.
- Use the PRDMAINT Domain to store packages for self-maintenance. The agent software uses this domain to heal and update itself.
- 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 HPCA, the PROFILE File appears after the first device has registered with the Configuration Server. This file contains information that is collected from managed devices. This information is used to connect to devices to deploy data managed by HPCA, and to see the configuration of the managed device.

The NOTIFY File contains information about attempts by the Notify function to update, remove or e-mail subscribers. This file appears after the first attempted Notify. For more information about Notify, see Chapter 7, Deploying Services.

HPCA Infrastructure

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

- management applications
- management infrastructure
- extended infrastructure
- management extensions

Some of the basic infrastructure components are described below. For more information on all of the HPCA products, see the *HP Client Automation Configuration Server, Porta,and Enterprise Manager Getting Started Guide* on the HP support web site.

Configuration Server

The Configuration Server is part of the management infrastructure, and resides on a single server or across a network of servers. Applications and information about the subscribers and agent devices are stored in the Configuration Server Database. The Configuration Server distributes packages based on policies established by the HPCA administrator. See the *Configuration Server Guide* for more information.

Portal

The Portal stores information about the target devices in your environment in it's Zone-based Directory Service, and provides the web-services to make these directory objects available for Client Automation jobs and management functions. The Portal is part of the HPCA extended infrastructure. Refer to the *HP Client Automation Portal Guide* for more information.

Proxy Server

If you want to reduce the load on the Configuration Server, or store your data closer to your agent devices, consider using a Proxy Server. The Proxy Server stores a copy of the data that are available to subscribers attached to the Proxy Server. The Proxy Server is also part of the extended infrastructure. Evaluate the potential benefits for each server and its attached subscribers individually. For more information, refer to the *Proxy Server Guide*.

Contact your HP representative for details on the Portal and the Proxy Server.

Administrator

HPCA comes with a set of tools used to carry out software management functions. You should become familiar with these tools. This is part of the management infrastructure. Refer to the *Administrator User Guide* for more information. These include:

- **Administrator Packager (Windows Only)**
  Use the Admin Packager to create groups of components, called packages, and promote them to the Configuration Server.

- **Configuration Server Database Editor (Windows Only)**
  Use the Configuration Server DB Editor (CSDB Editor) to view and to manipulate the Configuration Server Database.
- **Administrator Agent Explorer**
  Use the Administrator Agent Explorer to view and to manipulate HPCA objects on the agent device.
- **Administrator Screen Painter (Windows Only)**
  Use the Admin Screen Painter to create custom dialog boxes.
- **Administrator Publisher**
  Use the Administrator Publisher to create groups of components, called packages, and promote them to the Configuration Server Database.

The Administrator Publisher for Windows is used to publish Windows Installer files. Refer to the *Administrator User Guide* for more information

Management Applications

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

There are six types of management applications available for communicating with the Configuration Server. Install the only those agents for which you have obtained a license.

- **HP Client Automation Application Manager (Application Manager)**
  Use this agent to distribute mandatory applications throughout the enterprise. This agent is described in this book.
- **HP Client Automation Application Self-service Manager (Application Self-service Manager)**
  Subscribers install, remove, or update optional applications that are available to them in a service list. This agent is described in this book.
- **HP Client Automation Inventory Manager (Inventory Manager)**
  This agent 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.

- **The HP Client Automation Patch Manager (Patch Manager)**
  The Patch Manager analyzes and manages security patches. Refer to the *HP Client Automation Patch Manager Guide* (*Patch Manager Guide)*.
- **HP Client Automation OS Manager (OS Manager)**
  The OS Manager controls the provisioning of operating systems. Refer to the *HP Client Automation OS Manager Guide (OS Manager Guide)*.

If you install both the Application Self-service Manager and the Application Manager feature sets, you decide if an application is mandatory or optional, and specify who controls the installation of the application. Add the Inventory Manager to also find out the hardware and software configurations of the agent device.

Summary

- HPCA gives you the flexibility and control to efficiently manage desktop software.
- The Configuration Server 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 HPCA Agents

At the end of this chapter, you will:

- Understand the system requirements and permissions necessary to deploy the HPCA agents for Linux or Mac OS X.
- Be able to install the agents using either the graphical or non-graphical mode.

Install only the agents for which you have licenses. If you do not have a license, the agent will not authenticate with the Configuration Server.

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

System Requirements

- TCP/IP connection to a computer running Configuration Server.
- HPCA agent requires 20 MB free disk space.

Platform Support

For detailed information about supported platforms, see the release note document that accompanies this release.

Prerequisites

- We strongly recommend installing the agents as root (or with a Mac OS X user ID capable of issuing sudo commands). Root authority is required to apply owner and group designators to managed resources.
- Install the agent on a local file system.
- The installation program must be run from within Linux or Mac OS X. Although you can continue to work within your operating system (performing other tasks and operations) while the installation program is being executed, we strongly recommend that you do not.
- If you intend to run any of the graphical components of the agent software on a Linux operating system, make sure the Linux 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 2 [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, we urge you to relocate it before beginning installation. You will be prompted for this during the installation. If you choose to overwrite your existing agent, all your customized data will be lost.

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

To successfully run applications, standard Linux 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 Linux operating system paths for operating system executables and shared libraries. We recommend these be included as part of the logon scripts of the Linux user ID who installs, and will maintain the agents.

Table 3 Environment Variables

Platforms	Examples
Linux	`LD_LIBRARY_PATH=/lib:/usr/lib:$IDMSYS:$LD_LIBRARY_PATH` `PATH=/bin:/usr/bin:$IDMSYS:$PATH`

In Table 3 above, `$IDMSYS` represents the fully-qualified path to the agent 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 agent or HPCA Administrator graphical tools such as the Admin Publisher, the Admin Agent Explorer, and the presentation of the agent logon panel.

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

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

Red Hat Linux Prerequisites

The following library files are required to run the agent on Red Hat Linux on Itanium/Integrity systems:

- 32-bit emulator (ia32el)
- glibclibXaulibXdmcp
- libX11
- zlib

Install these files using the appropriate package for your operating system. For example, for RedHat Enterprize Linux 5, these can be installed using the following packages: `ia32el-1.6-13.el5.ia64.rpm, glibc-2.5-12.i686.rpm, libXau-1.0.1-3.1.i386.rpm, libXdmcp-1.0.1-2.1.i386.rpm, libX11-1.0.3-8.el5.i386.rpm and zlib-1.2.3-3.i386.rpm`.

Recommendations

- After you perform an installation, make sure the Application Manager is successfully connected to the Configuration Server. This registers the subscriber in the Configuration Server DB. 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 agent, consider the following:

- You can perform a local installation of the agents.
- Your HPCA systems administrator can perform a Remote Installation Setup. This process stores the installation media in a selected directory path. Later agent installations can be initiated from any number of intended agent 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 agent installation.
  - Ability to use a post-installation method, which runs any script or executable after the agent 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 agent will be connecting to.

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

Installation Methods

You can install the agents by:

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

Several parameters can be used on the command line when installing the agents. These parameters are used to install the agent using the graphical mode, non-graphical mode, plain mode, or silent mode. Table 4 below, describes the installation parameters.

Table 4 Command Line Installation Parameters

Parameter	Example	Description
`-mode plain`	`./install –mode plain`	Installs the agent 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 agent 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 Agent Installation

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

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

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

Installing the HPCA agent

This section describes both the graphical (using a GUI) and non-graphical (using a command line) installations of the agent.

Graphical Installation

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

Local Installation

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

To install the agent to a local computer using a GUI

These instructions will guide you through the local graphical installation of the agent. For the non-graphical installation instructions, see Non-graphical Installation on page 40.

1. At a command prompt, change your current working directory to the correct Agents platform subdirectory on the installation media.

   Example: For Linux, type: **`cd /dvdrom/Agents/linux`**

2. Type **`./install`**, (for Mac OS X, type **`sudo ./install`**) and then press **Enter**.
   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**.

3 Click **Next**. The End User License Agreement window opens.

4 Read the agreement and click **Accept** to continue. The Select Components to Install window opens.

5 Select the check boxes for the agents you want to install.

6 Click **Next**.

The next few steps are required for Linux operating systems only. If you are installing to a Mac OS X device, skip to step 11 and select the installation type.

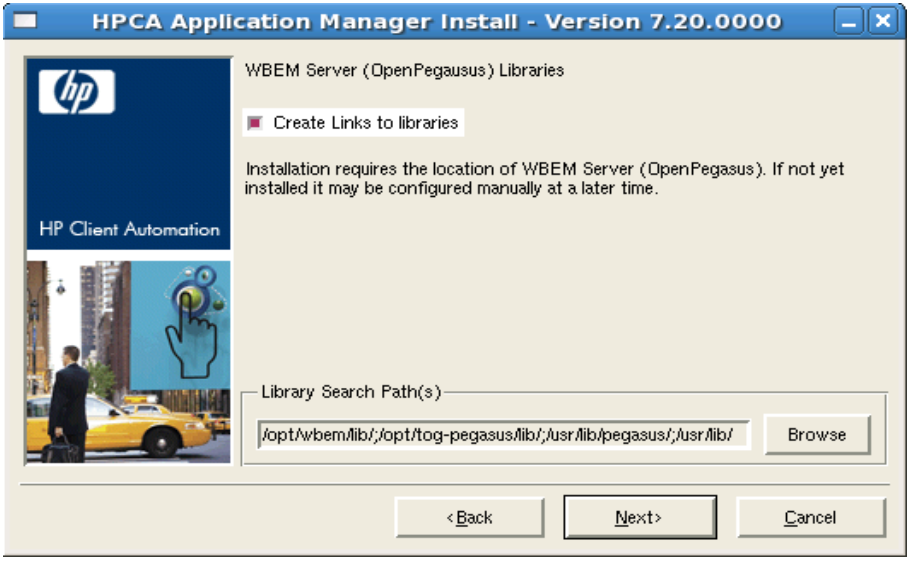
The Daemons window opens.

7 Select when you want the Daemons to start. The Daemons run on the client computer and perform HPCA management tasks. See About HPCA Daemons on page 46 for additional information.

— Select **Start after installation** to start the daemons after the Agent installation is complete.

— Select **Automatic start after reboot via init scripts** to configure the daemons to start automatically each time the device is restarted.

8 Click **Next**. The WBEM Server (OpenPegasus) Libraries window opens.



Select **Create Links to libraries** to create a link to existing WBEM Server libraries. Enter the location in the text box. Links can be created after the Agent is installed.

9 Click **Next**. The Select Installation Type window opens.

10 Select **Local Install** to install the agent onto a local computer, and then click **Next**. The Agent Location window opens.

11 Type the name of the directory where you want to install the agent, or click **Browse** to navigate to it.

12 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.

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

14 Click **Next**. The Log Directory window opens.

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

16 Click **Next**. The Configuration Server IP Address window opens.

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

18 Click **Next**. The Configuration Server Port Number window opens.

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

20 Click **Next**. The Package Settings window opens.

21 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.

22 When you are satisfied with the settings, click **Install** to install the agent with these settings.

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

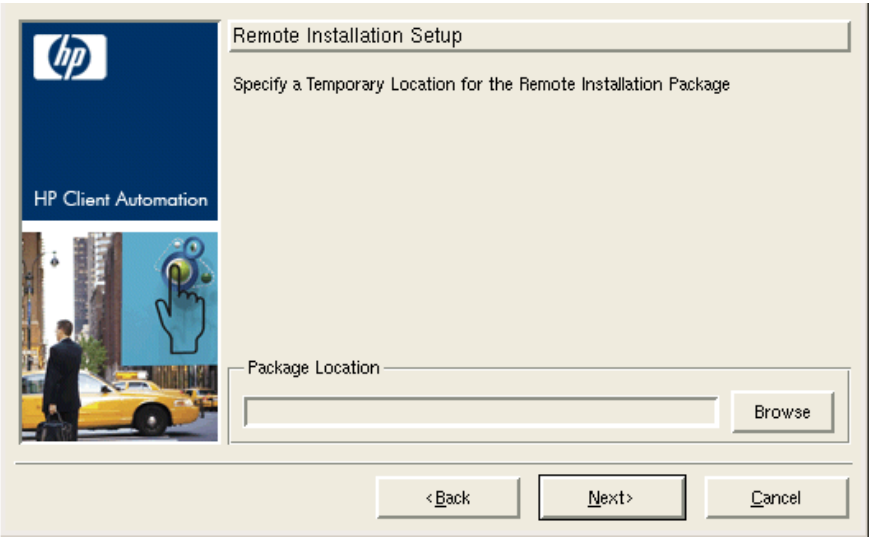
The agent has been successfully installed.

Remote Installation Setup

This section describes how to create an agent installation configuration file that can be used to install the Agent in silent mode or to a remote computer.

After the Remote Installation Setup is finished, a configuration file is saved in a directory you specify. Use the –cfg installation option to use the configuration file you created.

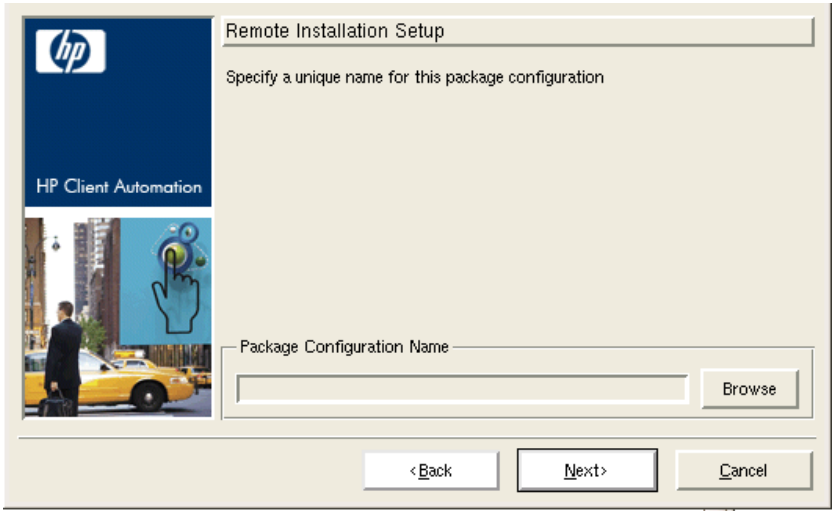
The remote installation is identical to the local install with the exception of two additional steps required for creating the remote installation package. Follow the steps for a local install, above, and when prompted, enter the required information for creating the remote installation package.



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

— Click **Next**.

The Package Configuration Name window opens.



— 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.

After a remote installation is complete, the agent installation media is 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 agent installations. These responses would normally be provided during an interactive agent 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 5, on page 37.

Table 5 Configuration File Variables

Variable	Sample Value	Description
REMOTE	0	0 designates a local installation. 1 designates a Remote Installation Setup.
INSTDIR	`/opt/HP/CM/Agent` Or `/Applications/HP/CM/Agent`	The default installation directory.
IDMLOG	`/opt/HP/CM/Agent/log` Or `/Applications/HP/CM/Agent/log`	This can be defined to designate a directory for IDMLOG other than the default `INSTDIR/log`.
IDMLIB	`/opt/HP/CM/Agent/lib` Or `/Applications/HP/CM/Agent/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	192.168.123.40	The default IP address for connection to the Configuration Server.
MGRPORT	3464	The default port number for connection to the Configuration Server.

Variable	Sample Value	Description
NTFYPORT	3465	The default Notify port used.
CONNECT	Y	Connects to the Configuration Server immediately after the installation. Default behavior is N. Set to Y if you want your agent to connect to the Configuration Server automatically after the installation.
OBJECTS	`./object.txt`	The file that is used to create or update HPCA 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 an agent. 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/HP/CM/Agent

# 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/HP/CM/Agent/radskman mname=NVDM,dname=SOFTWARE,ip=$ZIPADDR,
port=$ZDSTSOCK,cat=prompt,ind=y,uid=\$MACHINE,startdir=SYSTEM,ulo
gon=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 an HPCA Agent

We recommend that you install the agent as root.

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

- your HPCA 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 agent. Table 6 on page 40 describes these.

Table 6 Silent installation command line parameters

Parameter	Example	Description
`-cfg`	`./install –cfg install.cfg`	The file name 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 36.
`-mode silent`	`./install –mode silent –cfg install.cfg`	Installs the agent 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 36.

Non-graphical Installation

This section describes a non-graphical (using a command line) installation of the agent.

To install the HPCA agent using a command line

These instructions guide you through the local non-graphical installation of the agent. For the graphical installation, see Graphical Installation on page 32.

1. At a command prompt, change your current working directory to the correct operating system Agents subdirectory on the installation media.

   Example: For Linux, type: **`cd /cdrom/Agents/linux`**

2. Type **`./install –mode text`**,
   (for Mac OS X, type **`sudo ./install –mode text)`**
   and then press **Enter**. The agent installation begins.

3. Type **`C`**, and press **Enter**.

4. Read the license agreement, type **`Accept`** and press **Enter**.

5. In the next few steps, select which agents to install. Type Y or N and press **Enter** at each prompt.

   — Application Manager

   — Inventory Manager

   — Application Self-Service Manager

   — OS Manager

   — Patch Manager

   The next few steps are required for Linux operating systems only. If you are installing to a Mac OS X device, skip to step 9 and select the installation type.

6. You are prompted to start the Daemons after installation. Press **Enter** to accept the default (Y) and start the Daemons after install or type N and press **Enter** to start them later.

7. You are then prompted to automatically start the Daemons after a reboot via init scripts. Press **Enter** to accept the default (N) and **not** start the Daemons each time the device is restarted or type **`Y`** and press **Enter** to allow Daemons to automatically start when the device is rebooted.

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

   Type **`1`**, and then press **Enter** to install the agent locally.

   or

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

   For this example, we accepted the default.

9. Specify the installation location for the agent, and then press **Enter**.

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

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

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

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

14 Review the installation settings you have chosen.

15 If you would like to install the agent 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.

16 When you are satisfied with the settings, press **Enter** to install the agent.

   The agent is installed.

Installing the HPCA Agent to an HP Thin Client

With the HP Registration and Loading Facility (RALF) (see HPCA Registration and Agent Loading Facility on page 44) installed and registered with the HPCA infrastructure, you can deploy the HPCA agent to the thin client devices as you would normally (RALF is installed by default on the latest HP thin client devices.).

However, if you are installing the HPCA agent manually you will also need to install RALF (if it is not present already) after the Agent installation using the files provided on the HPCA media.

For RALF installations, “hpcaserver” or the host name defined using the RALF install parameters must be included in DNS. The host name of the HPCA Server must also be included in DNS when the Agent is installed from the HPCA console.

Installation of the HPCA Agent requires minimum free space of 5 MB on the /opt file system.

To manually install the Agent on a Linux-based thin client

1. Login to the target HP thin client device as root. If you are running ThinPro, you may have to create a custom connection for xterm (see note below).
2. Create a new directory called /opt/hpca.
3. Copy the install media from the appropriate Linux thin client sub-directory on the HPCA media to a temporary directory on the /tmp filesystem.
4. Change the working directory to the new temporary directory and run the installation by typing:

   `./install -i HPCAS_Server`

   Where *HPCA_Server* is the hostname or IP address of the HPCA Configuration Server.

   The HPCA Agent is installed.
5. If RALF is already present on the device, reboot the device when the Agent installation is complete.

   If RALF is not present, install RALF on the device. See To manually install the Registration and Loading Facility on Linux on page 45.

To remove the Agent from a Linux-based thin client

1. Login to the target HP thin client device as root.
2. Change the current directory to /opt/hpca/agent
3. Type ./uninstall and hit **Enter**.

   The Agent is removed.

To create a custom connection for xterm

If you are using the ThinPro operating system, you may need to create a custom connection to create an xterm connection.

1. From the HP menu in the lower left corner, select **Shutdown**.

2 From the Thin Client Action drop down, select **switch to admin mode** and specify the Administrator password (default password is root). Note: Control Center background will change from blue to red.

3 From the Control Center, click the **Add** drop down list and select the **custom** option.

4 Set Name to **xterm**.

5 Set Command to run to:

sudo xterm -e bash &.

6 Click **Finish**.

You now have a connection you can use to open an xterm session.

HPCA Registration and Agent Loading Facility

The HPCA Registration and Agent Loading Facility (RALF) is an agent component available for thin client devices managed by an HPCA Core infrastructure. RALF auto-registers the device with the HPCA infrastructure, and manages the HPCA agent install which is initiated from the main console. While RALF is part of the HPCA agent, RALF is available pre-installed on the HP thin client factory images, so registration can occur upon startup. If it is not on the factory image being used, RALF can be installed and configured on the gold image used for subsequent OS deployments. If installing RALF, the HPCA agent should also be installed prior to OS deployment.

RALF configuration and operation

RALF is shipped pre-installed on the latest HP thin client images (except those running ThinConnect). It is configured using a default HPCA Server hostname defined as 'hpcaserver.' While the HPCA server can be installed to match this name, it is more common to use this name as a DNS alias in defining the actual HPCA server host name. RALF can also be re-configured to define a different hostname using the command line options described below.

Once installed, RALF runs on Linux as a daemon that will periodically probe for the HPCA server. This probing will continue for 24 hours, and then RALF will shutdown. It will start this 24 hour probe again upon reboot. Once the server is contacted, RALF will register the device with the HPCA infrastructure and wait to accept the request to install the HPCA agent. Once

the agent is installed, RALF will periodically contact the server and verify device registration attributes.

Installation instructions for Linux (Debian or ThinPro)

You must have root authority to install RALF to Linux devices.

To manually install the Registration and Loading Facility on Linux:

1. On the HPCA Media, go to the *Media*\client\default\linuxtc\hpcaralf directory.
2. Copy the install media to /tmp on the Linux device.
3. Change the current directory to the /tmp directory.
4. On Debian devices:
   a. run dpkg -i hpcaralf.deb
5. On Thinpro devices (with read only root file system):
   a. Run fsunlock (to mount the file system as writable)
   b. Run /usr/share/hpkg/.hpkg_util -i hpcaralf.deb
   c. Run fslock (to remount the file system as read only)
6. After the installation is complete, either reboot the device or run /etc/init.d/hpcaralf to start and initialize RALF.

   You can use this script (/etc/init.d/hpcaralf) to start and stop the RALF daemon on the device.

RALF command line parameters

RALF supports the following command line options. These are here for documentation purposes, as most are used internally:

ralf.exe [-probe] [-host <host>] [-port <port>] [-debug] [-trace] [-version]

[-confinit]

[-help]

Table 7 RALF command line options

Option	Description
probe	Triggers the HPCA Probe

host	Specifies the optional HPCA Server host for probing and registration
port	Specifies the optional HPCA Server port for probing and registration
confinit	Defines the RALF Application configuration file entries for test environments
debug	Specify a Debugging logging level
trace	Specify a tracing logging level
version	Displays the Version of ralf.exe
help	Displays RALF information

About HPCA Daemons

The agent installation program installs the following daemon executables:

- **Notify (default port 3465)**
  Use 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.

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

The installation of **radexecd** and **radsched** as services on a Linux workstation is not automated within the context of the installation. The starting of services on Linux workstations is operating system dependent. For information about installing HPCA daemons as system services at boot

time, see your local Linux system administrator or refer to your Linux operating system's manual.

The installation of **radexecd** and **radsched** as services under Mac OS X will install sample startup scripts under Mac OS X POSIX style path `/Library/StartupItems/Radexecd` and `/Library/StartupItems/Radsched`

Sample Shell Scripts

The installation of the agent 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 Agent Installation

If you encounter any problems while installing the Agent, perform the following steps before contacting technical support:

- Enable diagnostic tracing by appending the text **`-loglevel 9`** to the installation command line and re-run the installation.
- Have this log file (`tmp/setup/setup.log`) located in the home directory of the 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 agents as root.
- The agents can be installed using either the graphical or non-graphical modes,

3 HPCA Agent Directories and Objects

At the end of this chapter, you will:

- Be familiar with key agent objects.
- Know the hardware attributes that the HP Client Automation agent collects.
- Understand the APPEVENT object.
- Know how Open Database Connectivity (ODBC) can help you generate reports with information from the objects.

This guide describes the *suggested* implementation for the HP Client Automation (HPCA) Agents, Application Manager and Application Self-service 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 these agents. This chapter describes agent objects.

Agent Directory Structure

Below is an example of the directory structure on an agent device. (Directories in Table 8 below are preceded with `/opt/HP/CM/Agent/` by default).

Table 8 Agent directories

Directory (/opt/HP/CM/Agent/..)	Description
`/opt/HP/CM/Agent`	Agent 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/CM-CS`	Configuration Server name (mname)
`../lib/SYSTEM/CM-CS/SOFTWARE`	Directory Name (dname)
`../lib/SYSTEM/CM-CS/SOFTWARE/ZSERVICE`	ZSERVICE Class
`../lib/SYSTEM/CM-CS/SOFTWARE/ZSERVICE/DRAGVIEW`	Sample application directory
`../log`	Log directory (IDMLOG)

Application Self-service Manager Directories

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

Table 9 NOVAEDM Parameters

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

About HPCA Agent Objects

When an agent device connects to the Configuration Server, information is exchanged between the agent and the Configuration Server. This exchange is called resolution. During resolution, HPCA checks the status of services, and updates the Configuration Server with information from objects stored on the agent device.

Agent objects are stored in the `IDMLIB` directory on the agent device. After installing the agent and connecting to the Configuration Server, you can use agent objects to answer questions such as:

- What is the hardware configuration of the agent device?

- Was the service successfully installed?
- When was the service installed?
- What is the agent device's name, and who was the last user logged on?
- What are the possible data sources for this agent device?

While there are multiple HPCA objects on an agent device at any time, there is a core group of objects that supply information about and the status of the current agent connect. Table 10 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 agent connect, but only the most commonly used ones are noted here. Check the HP support web site for information on other agent objects.

Table 10 Core Agent Objects

Object	When created or updated	Type of Information included
ZCONFIG	ZCONFIG is created at start of agent connect process. See Table 11 on page 55 for more information.	Contains basic hardware information for the agent device such as processor, operating system, and drives.
SYNOPSIS	This object is transferred to the Configuration Server at the end of the agent connect. Note: Client Operations Profiles must be enabled for this object to be present. See Table 12 on page 56 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.

Object	When created or updated	Type of Information included
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 13 on page 58 for more information.	The SAPSTATS object has one instance for each of the agent device's Server Access Profiles (SAP). It summarizes information for each SAP such as speed, number of files sent and received, and the role of the SAP.
PREFACE	PREFACE is sent to the Configuration Server at every phase of a radskman process including: • Client Operations Profile resolution • Self Maintenance resolution • Catalog resolution • Single service resolution (This can happen multiple times depending on what services are processed.) • Outbox flush See Table 14 on page 60 for more information.	Contains core information about each invocation of radskman.
SMINFO	(Linux Only) SMINFO is created at the start of the agent connect process. See Table 15 on page 61 for more information.	SMINFO collects information that is independent of the hardware and software installed on the device, and some network information.

Agent Version

To verify the agent's versions, open the connect.log file in the `IDMLOG` directory on any operating system using a text editor and search for "version."

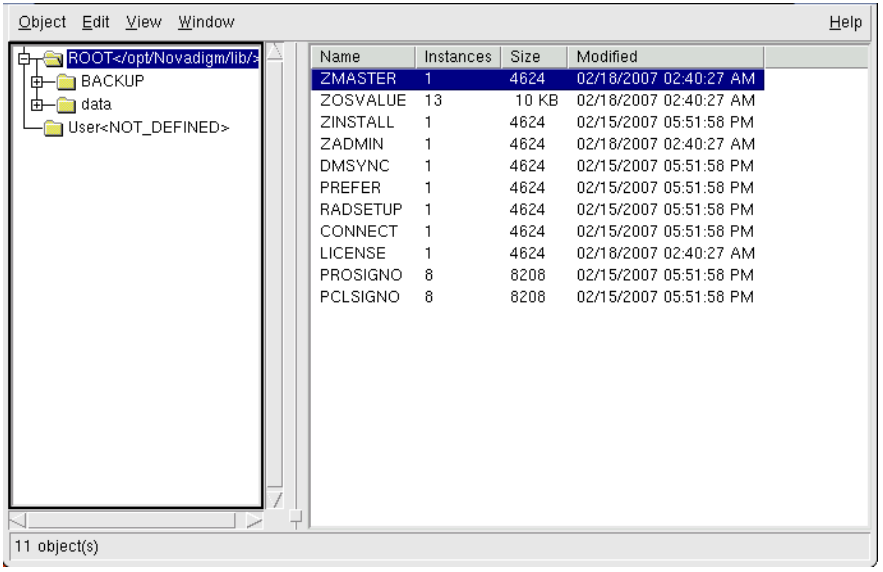
Using the Admin Agent Explorer to View Objects

The Admin Agent Explorer is installed as a component of the M Administrator. Use it to view objects in the `IDMLIB` directory. You can view any object if you have access to the agent device's `IDMLIB` directory. Otherwise, you may need to manually retrieve the object file, and store it on your Administrator computer.

To view an object using the Admin Agent Explorer

1. Start the Admin Agent Explorer by running `radobjed` in the directory where you installed the Administrator.

   The Admin Agent Explorer opens.



2. Navigate to the agent device's `IDMLIB` directory, or to the directory where the object is stored.
3. Double-click the object's name in the list view.

   The Admin Agent Explorer displays the selected object.
4. Click **Save/Exit** to close the dialog box.

Hardware Configuration Information (ZCONFIG)

The ZCONFIG object stores hardware configuration information from the agent device. Use the Admin Agent 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 agent device.

The ZCONFIG object is sent to the Configuration Server automatically for viewing with the Admin CSDB Editor. 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 agent device.

Table 11 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
ZOSMAJOR	Major Component of OS version	2
ZOSMINOR	Minor Component of OS version	4

Attribute	Description	Example
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 agents using Client Operations Profiles. The SYNOPSIS object summarizes the most recent agent connect. Use the SYNOPSIS object to confirm the success or failures of the agent connect process. See Chapter 5, Configuring Client Operations Profiles for more information on implementing Client Operations Profiles.

Table 12 SYNOPSIS object 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
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

Attribute	Description
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 agent device from where this was run
USEREXEC	User that executed the job
CMDLINE	Command line parameters used to execute this job

Service Access Profile Status (SAPSTATS)

The SAPSTATS object is generated on agents using Client Operation Profiles, and is used to report the Server Access Profile (SAP) status and usage statistics from the agent. 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 See Chapter 5, Configuring Client Operations Profiles.

Table 13 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 agent will connect to the SAP. Maintained by agent.
ROLE	Role of the SAP. Possible values are: **O**: Client Operations 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 agent device measured in Bytes per second

Attribute	Description
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 agent 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 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 object is sent to the Configuration Server at every phase of a radskman process including:

- Client Operations Profile resolution
- 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 agent 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 agent connect, the type of connect, and the context.

Table 14 PREFACE Object Attributes

Attribute	Description
CMDLINE	The `radskman` command line parameters used for the current agent 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 agent. The possible values are: RSM Application Self-service Manager RAM Application Manager RPS Proxy Server (for preloading application resources)
JOBID	The jobid specified on the command line for this connect (client versions 3.0 and above).
LOCALUID	The starting directory under IDMROOT on the agent device. 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	Agent device's machine name.
USEREXEC	The user who is currently logged on and who executed the command. For Notify and Timers, this would be SYSTEM. For logon scripts, this would be the subscriber's network account name.
ZCONTEXT	The value of ZCONTEXT as passed on the RADSKMAN command line. • **M** indicates that RADSKMAN was run in a machine context. • **U** indicates that RADSKMAN was run in a user context. • A blank indicates that no context was specified on the RADSKMAN command line; the context will default to the context in which the agent connect was launched.
ZDOMNAME	The Configuration Server 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.

Attribute	Description
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 agent. In most scenarios, this represents the machine name of the agent device, but may also contain the current user name or another value. The value found in this field is often used as the key for policy resolution or reporting. The UID `radskman` parameter sets this value.

Systems Management Information (SMINFO)

The Systems Management Information (SMINFO) object is created on all agent devices. The SMINFO object summarizes hardware specific information that is independent of what operating system or software is installed on the agent device. HPCA uses SMBIOS standards to access data about the BIOS. SMINFO also includes some network and user ID information.

Table 15 SMINFO object attributes

Attribute	Description
ASSETTAG	Unique Asset Tag number of the agent device from the BIOS.
COMPDOMN	Computer Domain
COMPNAME	Computer Name
IPADDR	Agent device's IP address
MACADDR	Agent device’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

Directories, objects, and log files created by HPCA are assigned permissions based on current umask settings and execute permissions on objects and log files are removed. In order to change the default permissions assigned when new directories, objects, and log files are created by HPCA 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/HP/CM/Agent/lib/SYSTEM/CMCS/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
  - For log files: IDMLOGPERM

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, log files, or directories are created by HPCA.
   - For directories: LIBPERM
   - For objects: OBJPERM
   - For log files: LOGPERM

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

   ```
   OBJPERM     0644
   ```

   To exclude write permissions for logs for group and other, create a DEFAULTS.EDM file with the following:

   ```
   LOGPERM     0644
   ```

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

```
LIBPERM     0777
```

The PROFILE File

Some agent objects such as ZCONFIG and ZMASTER are sent to the Configuration Server during an agent connect. The objects received from the agent device are stored in the PROFILE File in the Configuration Server Database (CSDB). Within the PROFILE File, each agent device is stored as a domain. By default, each agent device is identified by the subscriber who is currently logged on. The user 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 16 Objects in the PROFILE File

Instance	Information Recorded
ZCONFIG	Contains basic hardware information for the agent device such as processor, operating system, and drives.
ZMASTER	Contains information used to run the agents, such as user ID and operating system.
ZSVCSTAT	Contains information about the service after it has been successfully installed on the agent device. 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 agent connect, such as the number of objects going to and from the agent device.

Each domain contains several classes, which represent the objects received from the agent device. Use the Admin CSDB Editor to view the PROFILE File.

Reporting

The following components allow for reporting on agent data.

Reporting Server

As part of the extended infrastructure, the web-based Reporting Server allows you to query the combined data in existing Inventory Manager, Patch Manager, and Application Usage Manager databases and create detailed reports. In addition, you have the option of mounting an existing LDAP directory, which allows you to filter your data using your LDAP directory levels.

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

For additional information refer to the *HP Client Automation Reporting Server Installation and Configuration Guide*.

Messaging Server

The Messaging Server is a generic messaging service that can be used with many HPCA Infrastructure modules. Its job is to continually monitor a predefined data queue and dynamically route data objects to one or more external destinations. The Messaging Server provides retry, rerouting, and failover capabilities to ensure all data is transferred efficiently and reliably.

On a Configuration Server, the Messaging Server operates with the QMSG executable to handle the transfer of reporting data obtained from agents to the appropriate external databases.

For additional information refer to the *HP Client Automation Messaging Server Installation and Configuration Guide*.

Agent Logs

The agent has three primary modules: RADSKMAN, RADPINIT, and RADCONCT. However, the activity-reporting of these three modules is shared in one log file, `connect.log` (the default name).

The default location of `connect.log` is `/opt/HP/CM/Agent/log`.

When `connect.log` reaches 1 MB in size, a backup log (`connect.bak`) is created.

As stated, `connect.log` and `connect.bak` are the default names given to these logs. You can rename the log (using the parameter, **`log`**) in a format that better suits your needs. (For example, you might prefer to 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.) Additionally, you have the option of appending information to a log by using the parameter, **`alog`**. The **`log`** and **`alog`** parameters are discussed in Table 17 starting below.

Each of the three primary agent modules can be instructed to use a specific log file by simply adding the **`log`** parameter to its command line. The three primary agent modules take command-line parameters in the following format.

`Keyword` **`=`** *`value`* `(in comma-delimited format)`

Use the optional **`log`** and **`alog`** parameters on the command line to name the log file and append information to an existing log file, respectively. For example, you could add the **`log`** parameter to a RADSKMAN command line in a Notify in order to generate a specific log name, as in:

```
radskman log=notify10012003.log
```

Table 17 Parameters for Log Files

Parameter	Description
log	The name of the log file that is to be created, such as `Connect.log`, the default. Use a valid filename without a path (by default, logs are stored in the `IDMLOG` folder). If there is a log file with the same name, HPCA creates a backup of that file called *`logname`*`.bak`. If there is an existing *`logname`*`.bak`, it will be overwritten.

Parameter	Description
alog	The name of the log file to which the information will be appended. For example, **`alog=Application1.log`**. Use a valid filename without a path (by default, logs are stored in the `IDMLOG` folder). This parameter has no default; if it is not specified, the information will be appended to the log file that is named in the **log** parameter.

The value for the **`log`** parameter is stored in the LOGNAME attribute, which is 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 agent. The information in the radstate output is based on data retrieved from numerous agent objects.

When `radstate` is run with the Verbose parameter (mode v), it provides basic information regarding the agent 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.
- Timer Information

Use radstate at anytime to check agent configurations. For example, run radstate at the end of each agent 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 agent-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 agent desktop. Radstate is executed from the command line using the appropriate parameters, separated by a comma, for example:

```
radstate mode=vo, IDMROOT=/opt/HP/CM/Agent/lib
```

See the HP support web site for a technical document with additional information.

Summary

- Use the Admin CSDB Editor or Admin Agent Explorer to review the information collected from an agent device. This information is stored in the PROFILE File.
- Use the APPEVENT object to see the status of application deployments.
- You can connect to an ODBC database to view objects or generate reports.

4 Implementing Entitlement Policy

At the end of this chapter, you will:

- Understand how Client Automation can integrate with your existing policy information.
- Understand the 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 describes the *suggested* implementation for the HP Client Automation Agents; Application Manager and Application Self-service 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 these agents. This chapter describes how to implement entitlement policy, assigning users to groups, and connecting applications to users.

About Policy Management

As your organization grows and changes, you may need to make changes to who has access to what software. With HPCA, you have the advantage of using your existing policy information, while using HPCA to manage your data.

HPCA 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 data are kept up-to-date.

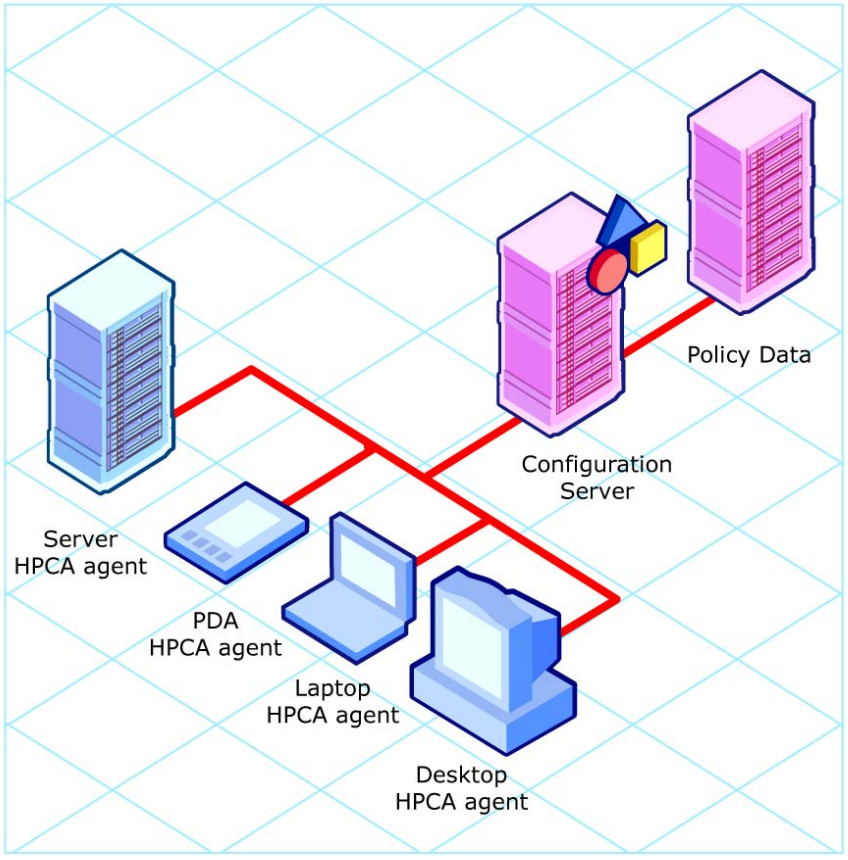
Accessing Existing External Policy Information

When an agent connects to the HP Client Automation Configuration Server, HPCA 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 HPCA to search the Configuration Server Database (CSDB) for this

information. However, typically, you will want HPCA to search your existing policy stores. This information is sent back to the Configuration Server, which determines which data are to be managed for the user, group, or computer.

For technical details about integrating your existing policy using the Policy Server, see the HP support web site and the *HP Client Automation Policy Server Installation and Configuration Guide (Policy Server Guide)*.

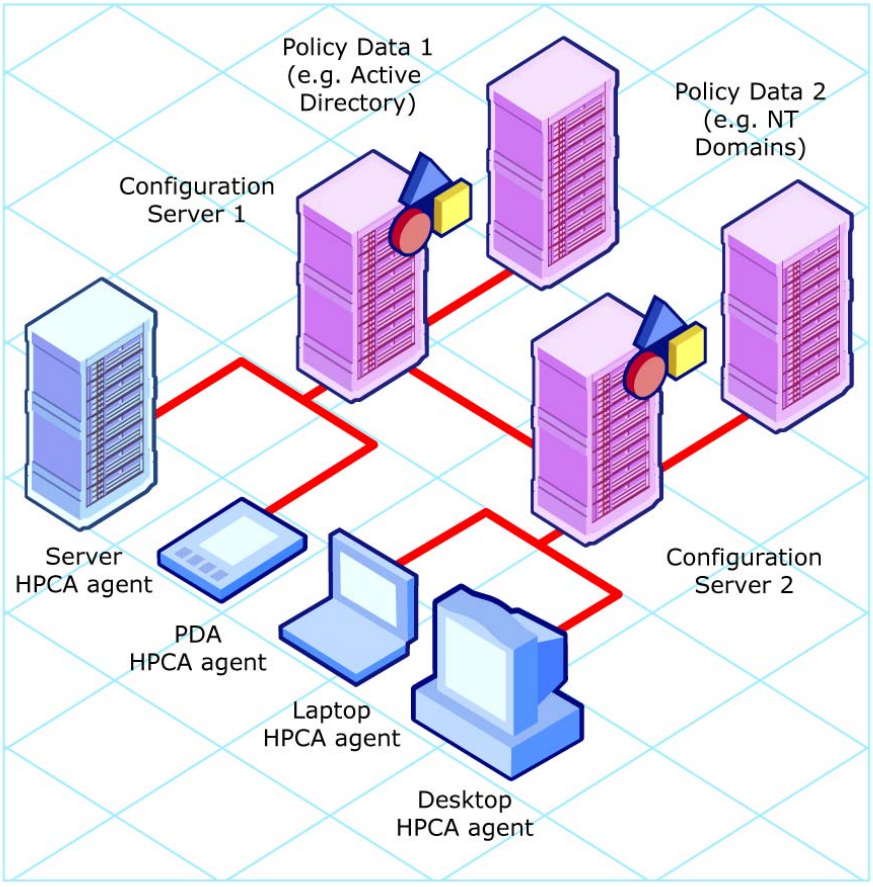
Figure 2 Policy information from an external source



HPCA 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 3 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 data, HPCA 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 HPCA, see the HP support web site.

Directories-Based Entitlement

(such as Active Directory and NDS)

Use the Policy Server to leverage your investment in LDAP-based directory services or SQL-based databases. The Policy Server is a plug-in to the HPCA Integration Server that is used for administration purposes such as mapping services to users in the directory tree. The Configuration Server can be configured to query the Policy Server to determine what services should be distributed and managed for the agent that is currently logged on.

The Policy Server is an optional feature available from HP. Contact your HP sales representative for details.

See the *Policy Server Guide* for more information.

HPCA integrates with existing policy, thereby greatly reducing the total cost of ownership of your environment by allowing you to continue to manage policies from your existing repository while HPCA manages your data.

About the POLICY Domain

If you are using real-time policy information from an external source to manage your data, you may need to configure a connection from your external policy store to the POLICY Domain in the CSDB. The configuration may vary based on the policy store.

This section provides 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 CSDB 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.

After you are familiar with the POLICY Domain and understand the basics of managing policy information within HPCA, you can extend that knowledge to learn how to integrate your existing policy information with HPCA. This information may also be useful if you want to create a simple lab environment to test the management of your data.

The following section uses the Admin CSDB Editor, which is available for 32-bit Windows platforms. For more information, refer to the *Administrator User Guide*.

To access the POLICY Domain

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.

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

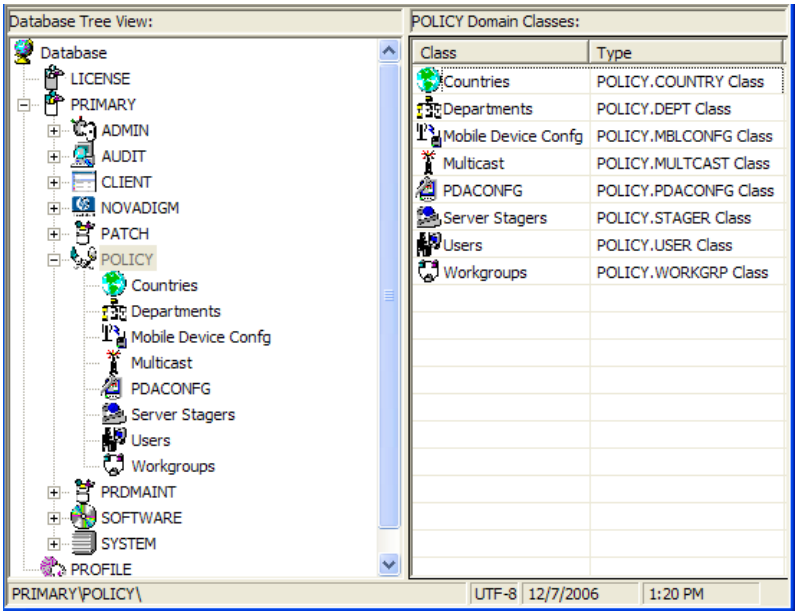
The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation.

You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

The Admin CSDB Editor window opens.

4 Double-click **PRIMARY**.

5 Double-click **POLICY**.



Classes in the POLICY Domain

The POLICY Domain has eigt default classes, Countries (COUNTRY), Departments (DEPT), Mobile Device Config (MBLCONFIG), Multicast (MULTCAST), PDACONFG (PDACONFG), Server Stagers (STAGER), Users (USER), and Workgroups (WORKGRP), as described in Table 18, below.

Table 18 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
Mobile Device Config (MBLCONFG)	Defines the parameters for mobile device configuration when using the Mobility Server.	RmmUser
Multicast (MULTCAST)	Use the MULTCAST class to configure agent devices to use multicasting.	MCast1, Mcast2

Class	Description	Instance Examples
PDACONFG (PDACONFG)	This class defines the parameters for PDA configuration.	PDAUser
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 composed 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 *Administrator User Guide* for information about creating new classes.

Creating Users or Groups

You may need to create individual users or groups in HPCA. For example, you might want to create a lab environment used to test the distribution and management of your data. To create a simple environment, 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 Configuration Server 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 Admin CSDB Editor to create a new user in the USER class.

The following instructions use the Admin CSDB Editor. The Admin CSDB Editor is currently available for 32-bit Windows platforms. For more information, refer to the *Admin UserGuide*.

To create a new user

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **CSDB Editor** → **Client Automation Admin CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.

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

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

You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

The Admin CSDB Editor window opens.

7 Double-click **PRIMARY**.

8 Double-click **POLICY**.

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

10 Select **New Instance**.

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

12 Click **OK**.

The user instance 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 Admin CSDB Editor to assign a user to the Sales department.

The Sales instance may not appear in your CSDB. To add this instance (or instances that are appropriate to your organization), follow the procedure To create a new user on page 77. However, instead of right-clicking USER, you would right-click the appropriate class, such as Departments (DEPT).

The following instructions use the Admin CSDB Editor. The Admin CSDB Editor is currently available for 32-bit Windows platforms. For more information, refer to the *Administrator User Guide*.

To assign a user to a department

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **Client Automation Administrator CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.

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

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

You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

The Admin CSDB Editor window opens.

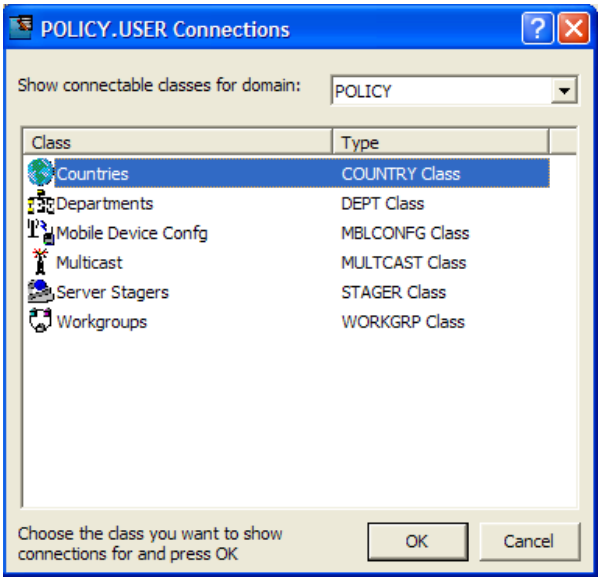
3 Double-click **PRIMARY**.

4 Double-click **POLICY**.

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

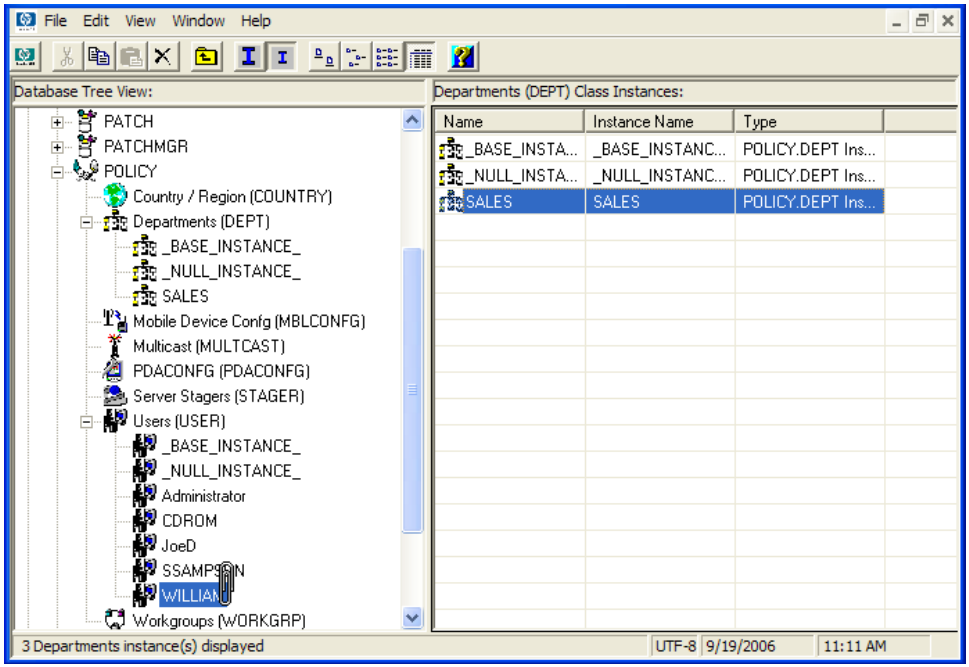
6 Right-click a user instance and select **Show Connections.**

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



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

8 Select the **Sales** instance from the list view and drag it to the appropriate Users instance.When your cursor turns into a paper clip, release the mouse button.



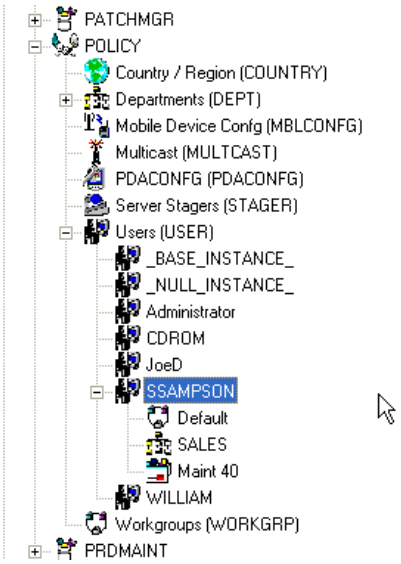
9 The Select Connection Attribute dialog box opens.

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

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

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

In the Admin CSDB Editor tree view, notice that Sales is now listed under the user instance, which indicates that user is part of the Sales department.



Connecting Services to Groups

Whether you are using an external policy source, or you are managing policy within HPCA, you will need to define the services that your subscribers will receive.

If you are using the Policy Server, refer to the *Policy Server Guide* for more information.

In this section, you will learn how to connect users and groups to the services that HPCA manages. In the following example, we will use the Admin CSDB Editor to authorize all subscribers in the Sales department for a sample application.

The following instructions use the Admin CSDB Editor. The Admin CSDB Editor is currently available for 32-bit Windows platforms. For more information, refer to the *Administrator User Guide*.

To connect an application to the Sales Department

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.

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

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

You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

The Admin CSDB Editor window opens.

3 Double-click **PRIMARY**.

4 Double-click **POLICY**.

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

6 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.

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

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

9 Select an application instance from the list view and then drag it to the appropriate Department instance. When your cursor turns into a paper clip (see the next figure), release the mouse button.

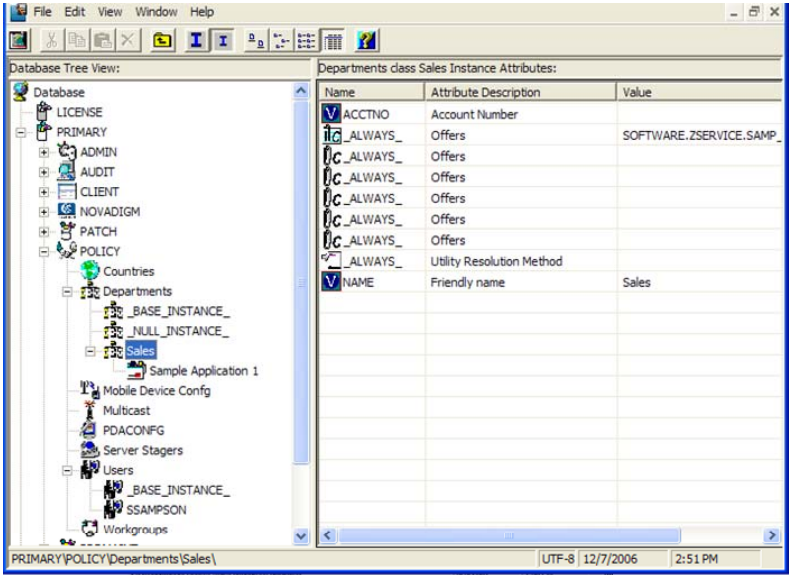
10 The Select Connection Attribute dialog box opens.

11 Click **Copy** to create the connection from Departments.Sales to the application.

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

13 Click **OK** when you receive the confirmation.

In the Admin CSDB Editor tree view (see the next figure), notice that the application is listed under the Sales department instance, which indicates that the entire Sales department is now authorized to receive the application.



Notice now that the user instance you created listed in the Users (USER) Class, is part of the Sales department. You can also see that the sample application has been authorized for the entire Sales department. Therefore, as long as the user SSampson is part of the Sales department, HPCA will manage the sample 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

- HPCA can integrate with your existing policy information.
- The POLICY Domain organizes subscribers into logical groups.
- You can create new users and assign them to groups.
- Assign the services managed by HPCA to the appropriate groups.

5 Configuring Client Operations Profiles

At the end of this chapter, you will:

- Understand the benefits of Client Operations Profiles (COPs).
- Know how to implement COPs.
- See a simple implementation example.

This guide describes the *suggested* implementation for the HP Client Automation (HPCA) Agents; Application Manager and Application Self-service 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 these agents. This chapter shows you to how to configure the operations of your agent. It describes the CLIENT Domain, and provides information on providing failover capabilities, designating servers for an agent based on criteria you set, controlling trouble shooting settings, hardware scan settings and user interface settings.

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 agent before connecting to the Configuration Server. Use Client Operations Profiles to do this. Client Operations Profiles, along with the HPCA object oriented schema, allow you to control certain agent behaviors based on any attribute from an agent object. Benefits of using Client Operations Profiles include, but are not limited to:

- Failover capability for HPCA servers.
- Dynamic assignment and selection of an agent device'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 Configuration Server Database version 3.1 or higher.

The CLIENT Domain

The CLIENT Domain in the Configuration Server Database (CSDB) controls COPs. It has six classes with sample instances that you can use to configure your agent device's operations. We provide an example of using the agent device's network location, stored in the ZCONFIG object, to prioritize the downloadable locations for application data for each agent device. 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 agent.
- **Hardware Scan Config (RADHWCFG)**
  Use an instance in this class to control the type of hardware scan that the agent should perform.
- **Network Location (LOCATION)**
  Use the LOCATION Class to group users based on a location, such as their subnet.
- **Notify Security Settings (NTFYSEC)**
  Use this class to use COPs to set RADEXECD to use internal authentication with the Management Portal.
- **RSM UI Preferences (RADUICFG)**
  Use instances in this class to manage the display of the Application Self-service 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 HPCA-managed services.

Recommendations

We make the following recommendations for using COPs.

- Use our Professional Services to help you implement this feature.
- Configure COPs only if you fully understand this process.
- Avoid single point of failure in all aspects, for servers of both types and for each role. Create redundancy where possible.
- Use base and null instances for unknown and new network addresses.

Implementing Client Operations Profiles

Use instances in the classes in the CLIENT Domain to customize the profiles to meet your enterprise's needs. There are five major steps for implementation.

1 Identify Servers.

2 Create Server Access Profile instances.

3 Set criteria for Server Access Profile instances.

4 Set priorities for Server Access Profile instances.

5 Enable Client Operations Profiles.

Each of these steps is detailed in the sections below. Before beginning this procedure, a discussion of Server Types and Roles is required.

Understanding Server Types and Roles

A Server Access Profile (SAP) is a generic way to define all possible data access points for a service. A SAP can be a Configuration Server, Proxy Server, or CD-ROM drive. COPs 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 agent 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 agent device's Client Operations Profile.
- **Service resolution (S)**
  Use this Configuration Server to resolve the agent device's services.
- **Agent self maintenance (M)**
  Use this Configuration Server to perform agent self-maintenance.

- **Reporting (R)**
  Use this Configuration Server for storing reporting objects from the agent device. These objects are stored in the PROFILE File in the Configuration Server Database.
- **Data download (D)**
  Use this Configuration Server to download application data to the agent device.
- **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 an agentis ready to download files, the agent 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 agent is still unable to download all the needed files after processing all servers with TYPE=DATA, then the agent 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 agent device must be set to Y.

The SAP.ROLE for the Configuration Server must be either D or A.

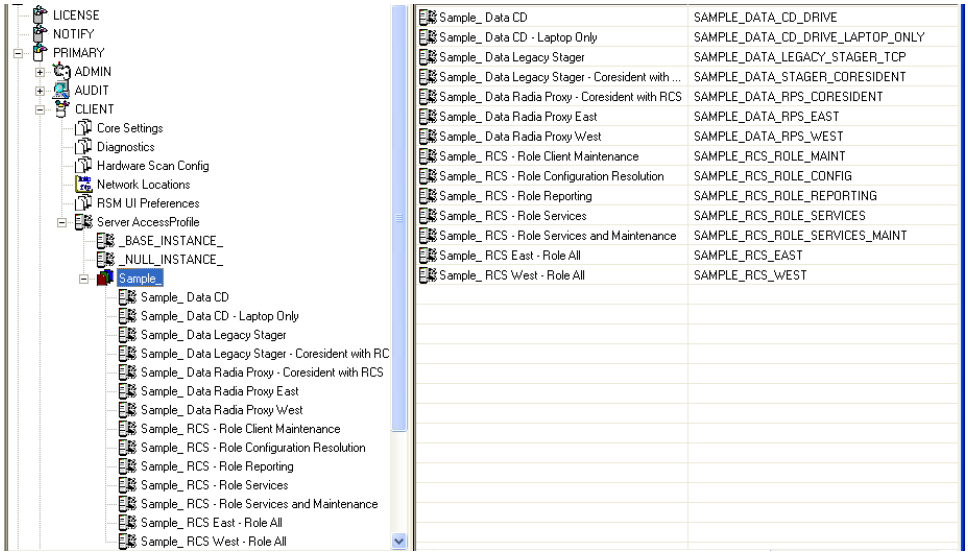
Task 1 Identify HPCA Servers

Identify your HPCA 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 HPCA Server can be a Configuration Server, Proxy Server, or CD-ROM. Servers with TYPE=DATA, must have a role of Data download (D). Configuration Servers that are 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 Admin CSDB Editor to create one SAP instance for each Server Access Profile. Table 19 below describes the attributes in the SAP class. We provide samples for each type of Server Access Profile in the Configuration Server Database. The next figure displays the SAP samples we provide.



Use the Admin CSDB Editor 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 19 belowto configure the instance for your enterprise.

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

Table 19 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.

Attributes	Description
TYPE	**Type [RCS /DATA]** Specify the Type of server. Set to `RCS` if using Configuration Server. Set to `Data` for Proxy Server or CD-ROM. If the agent device is unable to reach any of its Server Access Profile, then the agent 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.
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 = Agent 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.
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 agent timeout (ZMASTER.ZTIMEO) if it contains a valid numeric value. If blank, then use existing variable value on agent.

Attributes	Description
PUSHBACK	**Push Back (0-999 retries)** Set to 0 to skip a Configuration Server if the Configuration Server pushes back on the agent 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 agent bandwidth throttling if it contains a valid value. If blank, then use existing variable value on the agent device.
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 agent bandwidth setting if it contains a valid value. If blank, then use existing variable value on agent device.
STREAM	**Enable Streaming [Y/N]** Specify Y to use streaming. This will override the agent 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.

Attributes	Description
PROXY	**Internet Proxy URI** *Do not modify*. The Internet proxy URI through which the agent will connect to the SAP. Maintained by agent.
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 agents can use this SAP instance. Specify multiple agents separated by a comma. Below are suggested identifiers for each agent: Application Manager: HPCA-AM Inventory Manager: HPCA-IM Application Self-service Manager: HPCA-ASM OS Manager: HPCA-OSM Patch Manager: HPCA-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 HPCA-AM. Then, set product to HPCA-AM on your radskman command line. Default: Blank means that all products can use this SAP instance.
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 20, below.

Creating the Universal Resource Identifier

For each instance of the SAP class, you will need to identify a URI (Universal Resource Identifier) for the Server Access Profile (SAP). RFC 1630 proposes the following format:

scheme:*scheme specific format*

where the scheme is usually the network protocol such as HTTP or TCP. If the scheme-specific format has slashes, it indicates a hierarchical path. Universal Resource Locator (URL) is a form of URI where the scheme specific format is defined as:

//user:*password*@*host*:*port/url path*

Details on how to access the specified resource are defined in the URL path. The most prevalent form is: *//host*:*port/url path*. Examples for specifying the URI are show in Table 20, below:

Table 20 URI Example

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

Task 3 Set Criteria for each SAP Instance

After creating your SAP instances, you must decide how you are going to segment your enterprise. You may want to assign an SAP to an agent device based on its subnet. If so, use Admin CSDB Editor 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 an agent device includes an attribute called ZNETLOC. The ZCONFIG.ZNETLOC variable identifies the agent device's subnet using underscores instead of periods. You may want to name the Location instance based on possible subnets of your agent devices 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 99 shows you how to connect the subscribers to the appropriate location class using the agent device's ZCONFIG.ZNETLOC variable.

Task 4 Set Priority for each SAP for each Location

Use instances in the LOCATION Class to define the priorities of your Server Access Profile based on location criteria. The next figure 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.

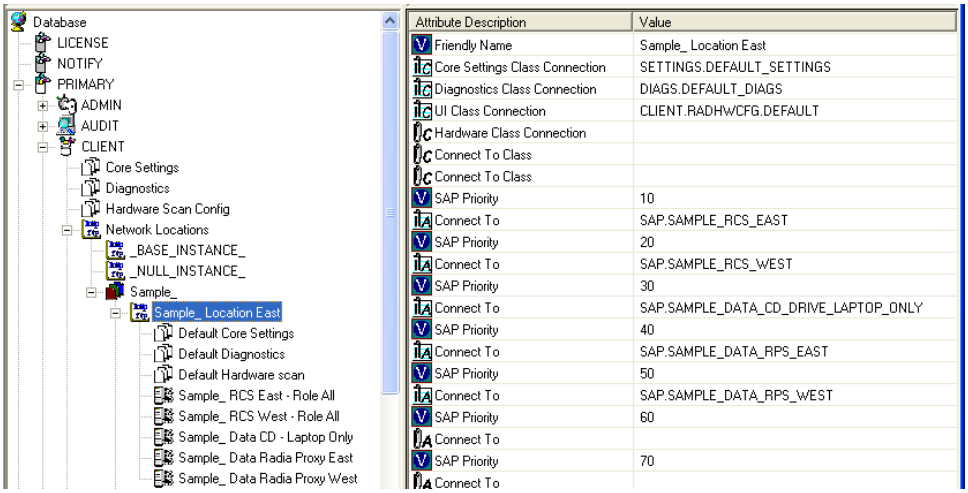


Table 21 below describes the attributes of an instance in the LOCATION Class. For example, if you want your SAMPLE_RCS_ WEST to be used before your SAMPLE_RCS_ EAST, increase its priority to 7. To do this, change the SAPPRI for SAMPLE_RCS_WEST to 7 as shown in the next figure.

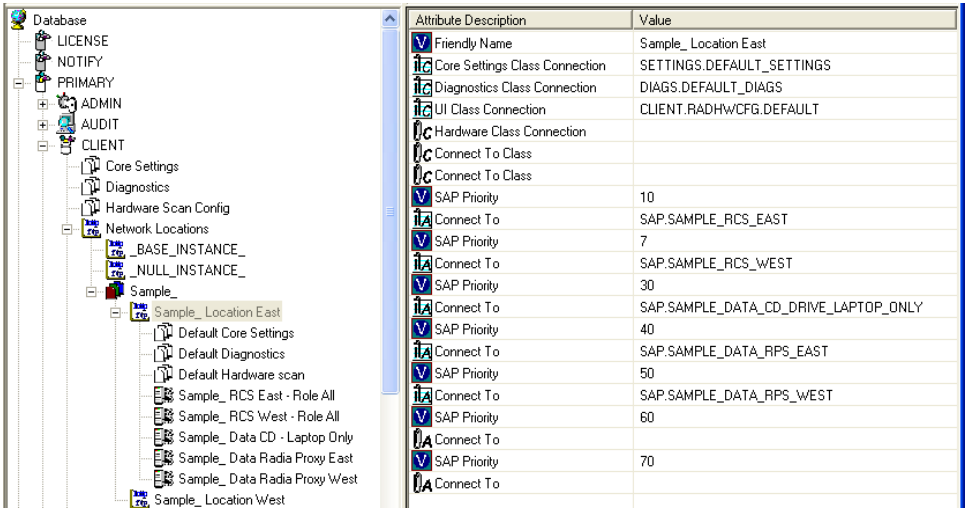


Table 21 Attributes of the LOCATION Class

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

Attribute	Description
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.
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.

Attribute	Description
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. Default: The SAP referenced in the connection below this instance has a priority of 30 by default.
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 30.
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. Default: The SAP referenced in the connection below this instance has a priority of 40 by default.
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 40.
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.

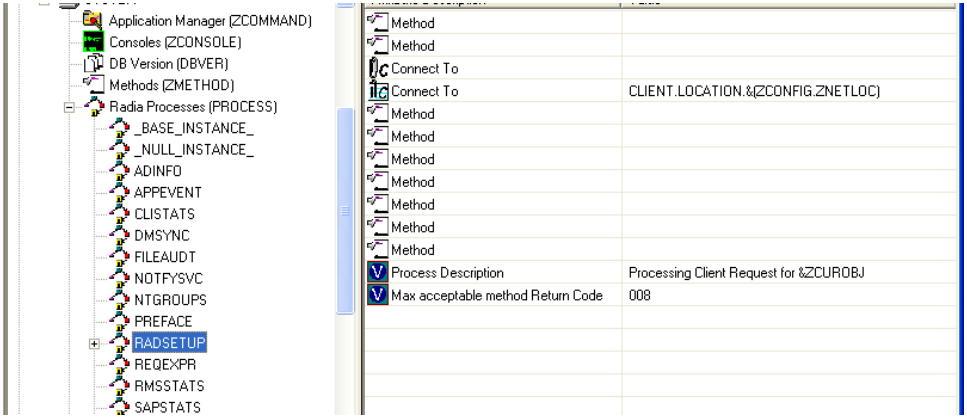
Attribute	Description
ALWAYS	**Connect to** Specify an SAP instance for the priority entered into SAPPRI above this attribute. This SAP will default to a priority of 60.
SAPPRI	**SAP Priority** Specify the priority of the SAP instance referenced in the _ALWAYS_ class connection below this attribute. 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 5 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 COPs are resolved. Second, you will need to enable the agent device to use COPs.

Enable on the Configuration Server

To enable COPs, you must create a new instance called RADSETUP in the 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 agent device's ZCONFIG object. The ZNETLOC attribute identifies the agent device'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 Agent

By default, COPs are disabled on agent devices for backwards compatibility with older version of HPCA. There are three ways to enable COPs on the agent device. Choose your method based on whether the agent has already been installed, and the method that suits your needs best.

If you have not already installed the agent,

- 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 COPs on already existing agents:

- You can customize the `install.ini` to add the COP variable to the RADSETUP object. To do this, add one line to the [objects] section in the install.ini. The figure below shows an example of the section with a new line.

  ```
  [Objects]
  ; Set CM object attribute values
  ; A value of _NONE_ will set the attribute to blank
  ;
  RADSETUP_COP=Y
  ;ZMASTER_ZDSTSOCK=
  ;ZMASTER_ZIPADDR=
  ;ZMASTER_ZNTFPORT=3465
  ```

If you want to enable COPes on already existing agents:

- 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 sample of code below shows the lines to add to your `initmeth.rex`. Be sure to deploy the updated `initmeth.rex`.

```
/* Sample INITMETH.REX to Enable the COP */
call edmget('RADSETUP',0)
RADSETUP.COP = 'Y'
call edmset 'RADSETUP'
```

- You can use the COP parameter of `radskman` to enable or disable COPs. To enable, add COP=Y to your list of parameters for `radskman`. This will only enable or disable COPs for this agent connect. Use `initmeth.rex` as shown above to enable COP for all agent 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 COPs after enabling it, run `radskman` with COP=N to disable for that agent 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 agent 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 22 Attributes of the SETTINGS Class

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

Attribute	Description
SAPPING	**Ping all SAP [Y/N]** Set to **Y** if the agent 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 agent connect. Set to 1 to 999 for number of retries if the Configuration Server pushes back. Default: The default setting is 0.
EQUISORT	**Secondary SAP Priority [R/S]** If several SAP instances have the same priority, set this to R to randomly select which one to use. Set to S to use the SAP with faster network speed. SAPPING must be set to Y to use EQUISORT= S. Use R for workload balancing. Default: The default setting is R.
USELSAP	**Use Last SAP [N/Y]** Set this Y to specify that the last SAP used in this agent 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 agent will go through the SAPs in priority for each service. During an agent 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.

Attribute	Description
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 agent devices 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 agent device's active directory information. The information is stored in the ADINFO object in the RADSETUP directory. The default location for the RADSETUP directory is the Agent lib directory. This information will be sent to the Configuration Server for all resolution processes. Default: The default setting is Y.
ZGRPINFO	(Windows Only) **Query NT User Group Info [Y/N]** Specify Y if you want to collect the agent device'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 the Agent lib directory. This information will be sent to the Configuration Server for all resolution processes. **Default**: The default setting is Y.

Attribute	Description
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 Agent on page 124. 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 agent object is corrupted for any reason, then it will be re-downloaded even if the desired state did not 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 does not change the desired state. Default: The default setting is Y.
ALWAYSS	**Always Upload CFG Objects [Y/N]** Set to Y to always upload all objects in the RADSETUP directory to the Configuration Server. The default location for the `RADSETUP` directory is the Agent lib directory. 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 agent device's `IDMSYS` directory. The default location is `/opt/HP/CM/Agent`. Default: The default setting is PRESETUP.REX.

Attribute	Description
EXASETUP	**Post Config Resolution Script** Specify a script to run after pre-configuration processing. This script must be in the agent device's IDMSYS directory. The default location is `/opt/HP/CM/Agent`.
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 agent device's `IDMSYS` directory. The default location is `/opt/HP/CM/Agent`. Note: COPs must be enabled on the agent for the EXBEXIT to be used. If COPs are not enabled, the EXBEXIT will be ignored.
TIMEOUT	**Communications Timeout (0-3200)s** Specify the timeout in seconds for the Server Access Profile (SAP). This will override the agent timeout (ZMASTER.ZTIMEO) if it contains a valid numeric value. If the value is blank, then the agent will use the existing timeout value on agent.
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.

Attribute	Description
BANDWDTH	**Bandwidth Percentage (1-99)** Specify the percentage of bandwidth to use between 1 and 99. If the value is blank or the variable does not exist, then all of the bandwidth will be used. This attribute applies to Windows only.
RADTRAY	(Windows Only) **Radtray Command Line Arguments** Set command line arguments you want to use for the System Tray. Specify Y for the first argument to enable the 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 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 System Tray in console mode and does not display the PAUSE button.
USEDEFS	**Use Default SAP [Y/N]** If a SAP cannot be found for the needed ROLE, specify Y to default to the 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.

Attribute	Description
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 agent 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 agent 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 HPCA to maintain. If a service is over the limit, it will not be installed.
REMUNINS	**Allow Remote Notify Uninstall [Y/N]** Specify Y to stops 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 agent 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 agent connect.

Attribute	Description
ACTMAINT	**Maintenance Activation [I/D/P]** The HPCA 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. Application Self-service Manager subscribers will receive a dialog box showing just an OK button that the Application Self-service Manager needs to be updated. Application Self-service 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 Application Self-service 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 Application Self-service Manager Interface. P is the same as I for Application Manager users. Default: I
SENDRPT	**Send Reporting Object [I/D]** Set to D to defer sending all reporting objects to Configuration Server at the end of agent 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

Attribute	Description
NETSPEED	**Speed Check Method [C/M/N/H]** Method used to check the speed of the connection. • **C** – (default) runs the ICMP check to the Configuration Server or Proxy Server. • **M** - runs the ICMP check to the Configuration Server or Proxy Server and also gets the speed of the network card, and returns the greater of the two values (for use when ICMP is disabled in the environment). • **H** - runs the ICMP check with a high performance counter to enable the ICMP check to differentiate between servers that have < 2ms response times. • **N** - turns off the network speed check.
NETTTL	**Network Time to Live (0-999)** Specify the number of *hops* (0–999) for the agent device to use for ICMP speed checks. The default is 3.
FLUSHU	**User Reporting Option [Y/N/A]** Supports flushing objects for all users during a connect or saving the objects locally for transfer at a later time. • **A** All Users • **Y** Always flush the outbox (default) • **N** Never flush the outbox On a user connect, a setting of FLUSHU=A will save off the reporting objects, which will then be transferred during a machine connect. On a machine connect, a setting of FLUSHU=A will transfer all (machine connect AND user connect) reporting objects. On a user connect, a setting of FLUSHU=N will on user connects to build up reporting objects in each user's outbox folder. Note: FLUSHU=N is applicable only for user connects and cannot be used for machine connect. During a machine connect specify FLUSHU=A to transfer all objects. The default (Y), will always send the current connecting client's reporting objects from the outbox folder.

Diagnostics (DIAGS)

Use this class to override default trace settings on the agent device. 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 agent. The information in the radstate output is based on data retrieved from numerous agent objects. For additional information on radstate, see the HP support web site.

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 agent device 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 23 Attributes of the DIAGS Class

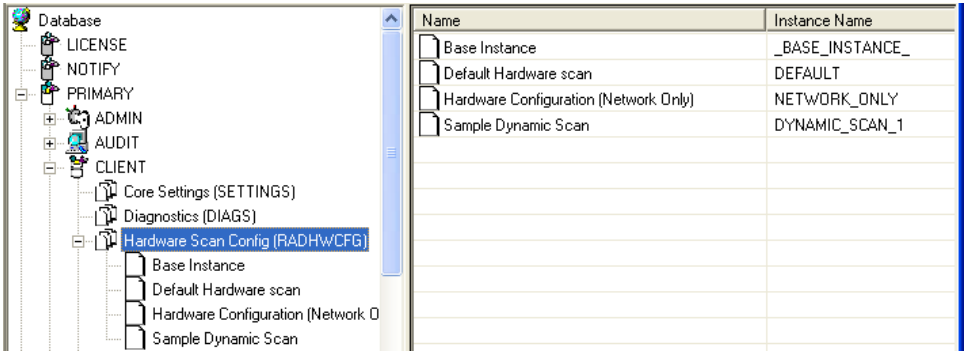
Attribute	Description
NAME	**Friendly Name** Type the friendly name of the instance.
RADSTATE	**Command String for radstate** This will run radstate with the parameters specified in this attribute. The base instance of the DIAGS Class is set to VO, which will run radstate in verbose mode, building the ZRSTATE and ZRSTATES objects. See the documentation on radstate for more information. If no parameters are specified, radstate will not run. Radstate must exist in the `IDMSYS` directory. You only need to specify the parameters for radstate, not the radstate executable. Refer to the document on radstate for additional information.

Attribute	Description
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 agent 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 agent log. All data transmitted and received will be echoed to the agent log file Caution**:** Setting ZTRACE = Y may generate very large logs and severely impact performance of the agent. 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 agent. 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 agent 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 agent 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.



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.

Table 24 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 agent device has only one printer connected, then only one ZHDWPA0*n* attribute will be reported in ZCONFIG.

Table 24 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

Attribute	Description
OS	**OS [Y/N]** Specify Y to scan for Operating System information. ZCONFIG attributes: REBOOTD, REBOOTT, WTSSRVR, ZHDWLANG, ZHDWOS, ZHDWOSDB, ZHDWOSOG, ZHDWOSOW, ZHDWSVCP
MEMORY	**Memory [Y/N]** Specify Y to scan for memory information. ZCONFIG attributes: ZHDWMEM, ZHDWMEMF
HDLOCAL	**Local Drives [Y/N]** Specify Y to scan for internal hard drives. ZCONFIG attributes: ZHDWCDDR, ZHDWD00, ZHDW00C, ZHDWD00F, ZHDWD00S, ZHDW00T, ZHDWD01, ZHDW01C, ZHDWDF_A, ZHDWDLST, ZHDWDNUM
HDREMOTE	**Remote Drives [Y/N]** Specify Y to scan for external hard drives. ZCONFIG attributes: ZHDW00, ZHDWD00C, ZHDWD00F, ZHDW00S, ZHDW00T, ZHDWDLST, ZHDWDNUM
NETWORK	**Network [Y/N]** Specify Y to scan for network information. ZCONFIG attributes: GATEWY01, IPADDR01, LADAPT01, NETLOC01, SUBNET01, ZGATEWAY, ZHDWIPAD, ZHDWLANA, ZHDWNET1, ZHDWNNET, ZNETLOC, ZSUBNET
PERIPHER	**Peripherals [Y/N]** Specify Y to scan for peripherals such as keyboard and mouse. ZCONFIG attributes: ZHDWKYBD, ZHDWMOUS, ZHDWPPAR, ZHDWPSER, ZHDWVIDO, ZHDWVRES
PRINTER	**Printers [Y/N]** Specify Y to scan for printers. ZCONFIG attributes: ZHDWPA00, ZHDWPA01, ZHDWPPRN

Attribute	Description
HAL_VER	**HAL Statistics [Y/N]** Specify Y to scan for the HAL (Hardware Abstraction Layer) version. ZCONFIG attributes: HALCOMP, HALDATE, HALFNAME, HALFVER, HALINAME, HALLANG, HALPNAME, HALPVER, HALSIZE.
APP_VER	**Application Version [Y/N]** Specify Y to scan for versions of MSI (ZHDWVMSI) and IE (ZHDWVIE).
WMISCAN	**Use WMI to collect data [Y/N]** Specify Y to perform the scan using WMI (Windows Management Instrumentation).
DSCAN00n	**Dynamic Scan 00n** Specify Y to use the dynamic scan variable.

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 Linux version of HPCA. 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 Parmn 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.

Notify Security (NTFYSEC)

Use this class to use COPs to set RADEXECD to use internal authentication with the HPCA Portal.

- This class contains a DEFAULT_NTFYSEC Instance that inherits all values from the _BASE_INSTANCE_ of the class.

  By default, this instance is connected to the **SAPPRI** (**SAP Priority 30**) Attribute of **PRIMARY.CLIENT.LOCATION._BASE_INSTANCE_**.

Table 25 Attributes of the NTFYSEC _BASE_INSTANCE_

Attribute	Description
ZNTFYSEC	Enable the notify-security feature. Valid values are **Y** (Yes), **N** (No), and **L** (Lock). The default is Y.
ZVRFYUID	Specify whether to verify the internal user ID. Valid values are **Y** (Yes), **N** (No), and **I** (Internal). The default is N. Note: In PRIMARY.CLIENT.LOCATION._BASE_INSTANCE_, the default value for this attribute is I.
ZVRFYPWD	Specify whether to verify the internal password. Valid values are **Y** (Yes), **N** (No), and **I** (Internal). The default is N. Note: In PRIMARY.CLIENT.LOCATION._BASE_INSTANCE_, the default value for this attribute is I.
ZEXTSEC	Enable HPCA *extended security*. Valid values are **Y** (Yes) and **N** (No). The default is N.
ZIGNRURI	This anti-spoofing attribute lets you to enable RCSURI stripping (the notify daemon strips out the value before executing the command). Valid values are **Y** (Yes) and **N** (No). The default is N.
ZIGNDURI	This anti-spoofing attribute lets you to enable DATAURI stripping (the notify daemon strips out the value before executing the command). Valid values are **Y** (Yes) and **N** (No). The default is N.

Usage Note

Changes to the notify-security settings will not be immediately effective; an HPCA agent connect must be performed in order to enable the new settings. This differs from other COPs settings, which happen as part of a single connect.

Additional Documentation

The following HP support documents contain additional information about notify security, the notify daemon, RCSURI and DATAURI stripping, and NTFYSEC. They can be found on the HP support Web site.

- *Radia Client-Notify Security*
- *Configuring an EDM:Client to Receive Notify Messages*
- *Enhanced Notify Security for Configuration Manager Agent v5.x*

Setting User Interface Properties (RADUICFG)

Use the RADUICFG Class to specify settings for the Application Self-service Manager User Interface. You must be licensed for the agent. To implement the hardware scan options, connect an instance of the RADUICFG Class to an instance in the LOCATION Class.

You must be licensed for and install the Application Self-service Manager to use this class.

Table 26 Attributes of the RADUICFG Class

Attribute	Description
BNHOME	**Display Home Button [Y/N]** Specify Y to display the Home button on the HPCA Bar.
BNMYSOFT	**Display My Software [Y/N]** Specify Y to display the My Software button on the HPCA Bar.
BNPREFER	**Display Preferences [Y/N]** Specify Y to display the Preferences button on the HPCA Bar.
BNHISTRY	**Display History [Y/N]** Specify Y to display the History button on the HPCA Bar.
SHWMENUS	**Show Menus [Y/N/U]** Specify U to allow the user to control appearance of the menu bar. Specify Y or N to turn the Menu bar on or off, and not allow the user to control its appearance.

Attribute	Description
SHWCATLG	**Show Catalog [Y/N/U]** Specify U to allow the user to control appearance of the list of catalogs. Specify Y or N to turn the catalog list on or off, and not allow the user to control its appearance.
STRTCHNG	**Startup param modifications [Y/N]** Specify Y to allow the user to modify the startup parameters in General Options in Preferences. Specify N to not allow the user to modify the startup parameters.
STRTFILE	**Startup parameter filename** Specify the file name for the startup parameters found in the General Options in Preferences.
STRUPMSG	**Warn if Startup file changes [Y/N]** Specify Y to warn the user if the startup parameter file has changed.
ASKOFFL	**Prompt for offline mode [Y/N/U]** Specify U to allow the user to control prompting for offline use of Application Self-service Manager. Specify Y or N to turn the prompt on or off, and not allow the user to control the prompt.
STATSTRT	**Status window on startup [Y/N]** Set to Y to show the status window on start up.
CUSTTTLE	**Custom Title on RSM Banner** Specify the text to display in the Application Self-service Manager's title bar.
COLTYPE	**Columns [Forced/Required]** Set to `Forced` if you want only the columns specified in COLNAMES to appear. Set to `Required` if at least the columns specified in COLNAMES should appear. Name and Status are always displayed.
COLNAMES	**Name of Columns (CSV format)** Specify the columns you want displayed. Separate the columns with a comma.

Attribute	Description
EXPSITEM	**Expand Active Service item [Y/N/U]** Specify U to allow the user to control the expansion of the active Service List item. Specify Y or N to allow or not allow the user to expand the active item in the Service List.
SHWADVOP	**Show Advanced Options [Y/N/U]** Specify U to allow the user to control the display of Advanced Options. Specify Y or N to turn off or on the display of Advanced Options such as **Download Only**, **Reconfigure**, and **Undo** buttons.
BTNINST	**Enable Install Button [Y/N]** Specify Y to enable the **Install** button.
BTNUPDT	**Enable Update Button [Y/N]** Specify Y to enable the **Update** button.
BTNDWLD	**Enable Download Button [Y/N]** Specify Y to enable the **Download** button.
BTNVRFY	**Enable Verify Button [Y/N]** Specify Y to enable the **Verify** button.
BTNREPR	**Enable Repair Button [Y/N]** Specify Y to enable the **Repair** button.
BTNDEL	**Enable Delete Button [Y/N]** Specify Y to enable the **Delete** button.
BTNCANCL	**Enable Cancel Button [Y/N]** Specify Y to enable the **Cancel** button.
BTNPAUSE	**Enable Pause Button [Y/N]** Specify Y to enable the **Pause** button.
SHWCOLEX	**Show Expand/Collapse Button [Y/N]** Specify Y to show the **Install** button .
SHWINFO	**Show/Hide Extended Info [Y/N]** Specify Y to show the **Extended Info** button when a service item is expanded.

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

Client Operations Profile Example

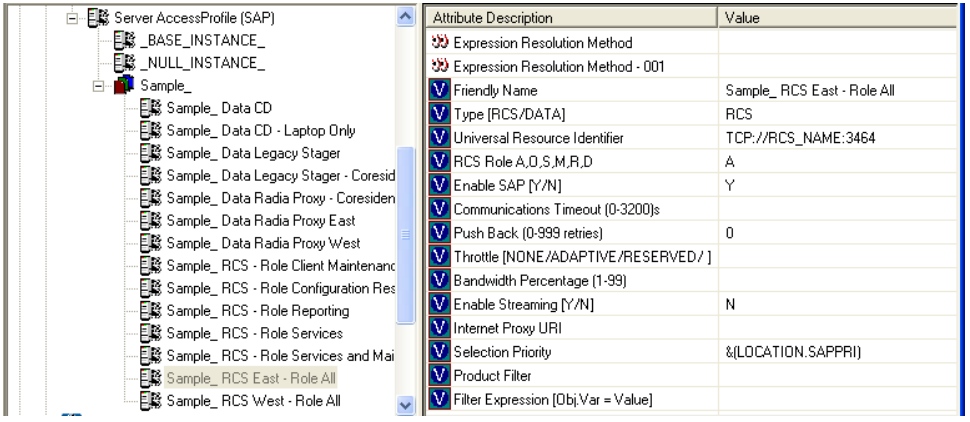
This section provides a simple example of how to configure COPs. The driving force is to have your agent devices connect with the most appropriate Configuration Server. Usually, you will want to assign your agent devices to a Configuration Server based on network address.

Scenario

Suppose you divide your enterprise into two regions, EAST and WEST. All agent devices in the EAST region are in the 192.111.111.0 network, and all agent devices 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 HPCA server for the EAST region, and one called Configuration RCS_WEST as the primary 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 next figure shows a SAMPLE_RCS_EAST.



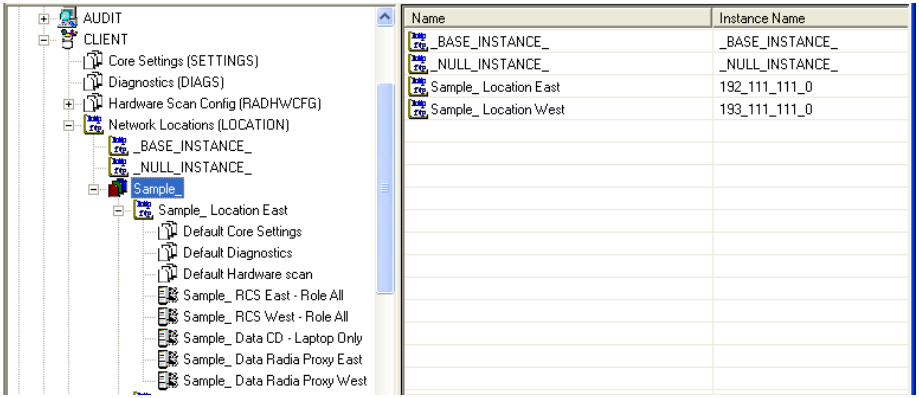
In the Server Access Profile (SAP) class use the TYPE attribute to specify a server as type DATA or RCS. In this example, we will be configuring only Configuration Servers. Therefore, all servers will have SAP.TYPE set to RCS.

For each Server Access Profile instance, you must also identify a role. Again, for simplicity, we will set SAP.ROLE to A for all. This means that the Configuration Server can provide agent operations profiles, service resolution, maintenance, data, and reporting.

At a minimum, you will need to specify the Universal Resource Identifier (URI) attribute. Customize other variables as needed.

2 Build two location instances, one for the EAST region, and one for the WEST region.

Create a location instance called 192_111_111_0 with a friendly name of Sample_Location East, and a location instance called 193_111_111_0 with a friendly name of Sample_Location West. See the figure below for an example.



3 Connect the LOCATION instance to the appropriate Server Access Profile (SAP) instance.

— In the LOCATION.Sample_Location East instance, define a connection to the SAP.Sample_RCS EAST.

— In the LOCATION.Sample_Location West instance, define a connection to the SAP.Sample_RCS WEST.

4 Now, consider what you want to happen in the following cases:

— Suppose you are an agent 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 an agent 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.

After 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 COPs to provide redundancy in your environment.
- Select which servers will perform which roles.
- You can assign agent devices to specific servers based on network location or any other criteria.
- You must enable COPs in the CSDB and on the agent device.

6 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 agent device.
- Know how to install services under the system account.
- Be familiar with preparing versioned applications.

This guide describes the *suggested* implementation for the HP Client Automation Agents; Application Manager and Application Self-service 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 these agents. This chapter covers preparing services.

Restarting the Agent Device

You may need to restart an agent device 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 agent device 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 27 below 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 27 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.

Application Events	Code	Description
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 VA 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. HPCA 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 agent device 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 Application Self-service Manager and for Application Manager used with the System Tray. If you do not want to show a warning message, specify ask=N in a radskman command line.

- **Quiet (Q)**
  No reboot panel will be displayed.
- **OK Button (A)**
  A warning message will display with an OK button only. Click **OK** to initiate the reboot. The user will not be able to cancel the restart.
- **OK and Cancel Button (Y)**
  Click **OK** to 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 agent 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. HPCA 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 agent 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

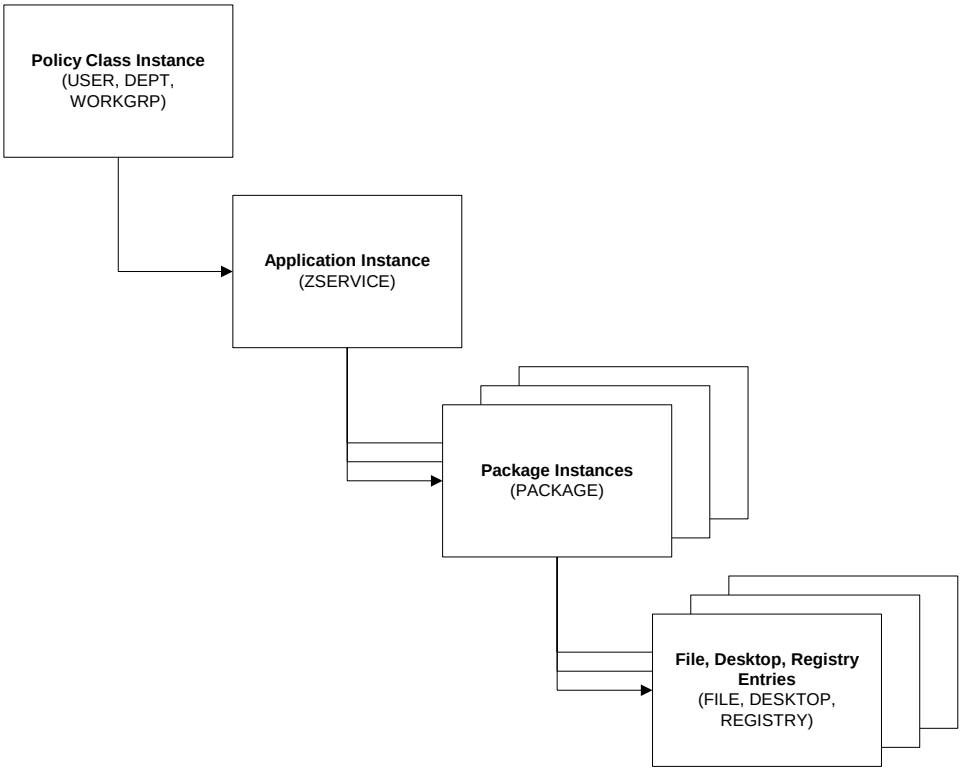
Normally, when you deploy an application to an agent device, it is activated immediately. This is the case when you use Scheduler or Notify. However, you can use Version Groups to roll out a new version of an application to subscribers, and activate it upon delivery or at a pre-determined time. If the installation of the new version fails, HPCA 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 agent device, 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 Configuration Server Database (CSDB). For non-versioned applications, one application instance connects to one or more package instances.

Figure 4 Model for non-versioned deployments

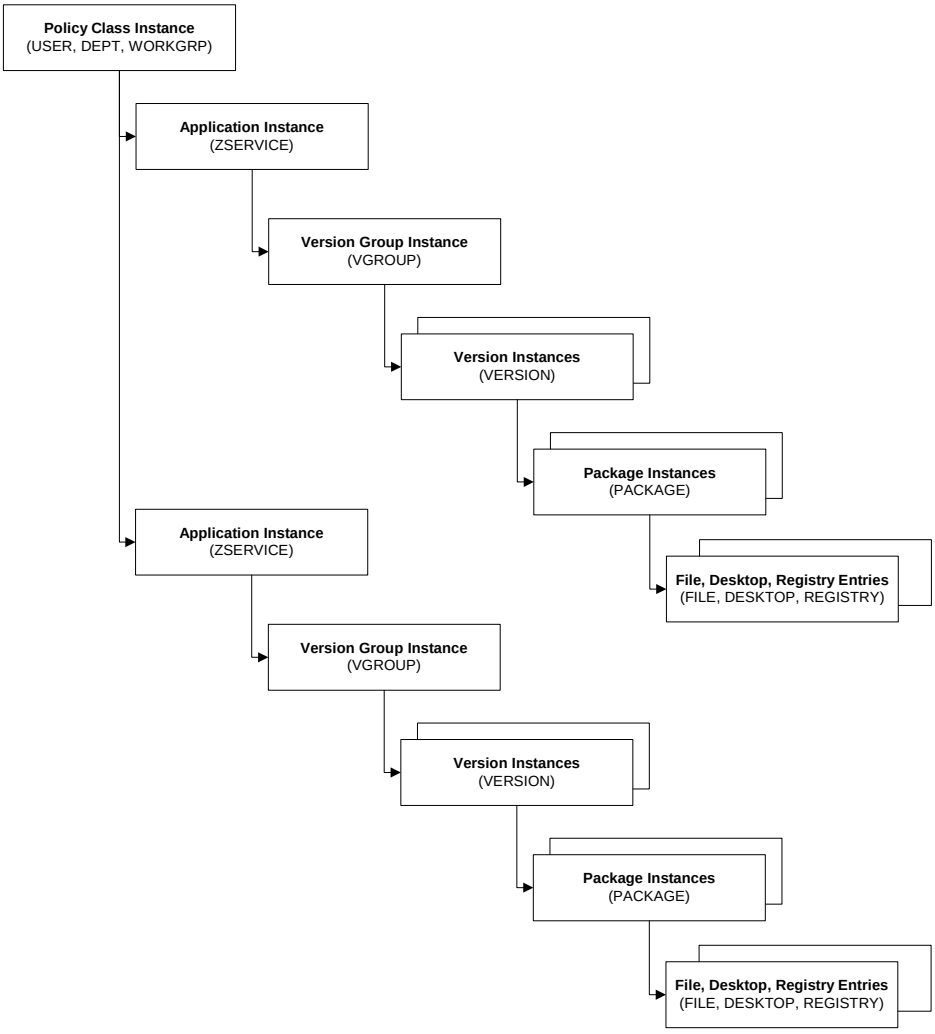


Versioned applications adhere to a different connection model than non-versioned applications. For versioned applications, an Application instance (ZSERVICE) connects to a single Version Group (VGROUP) instance.

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 HPCA package. Each package is represented in the CSDB by an instance of the PACKAGE Class.

Figure 5 Model for versioned deployment



To prepare versioned applications

1. Use the Administrator 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 Admin CSDB Editor, 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 Admin CSDB Editor to create a new instance in the Version Group (VGROUP) class.

To create a Version Group

1. Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**.

   The Admin CSDB Editor Security Information dialog box opens.

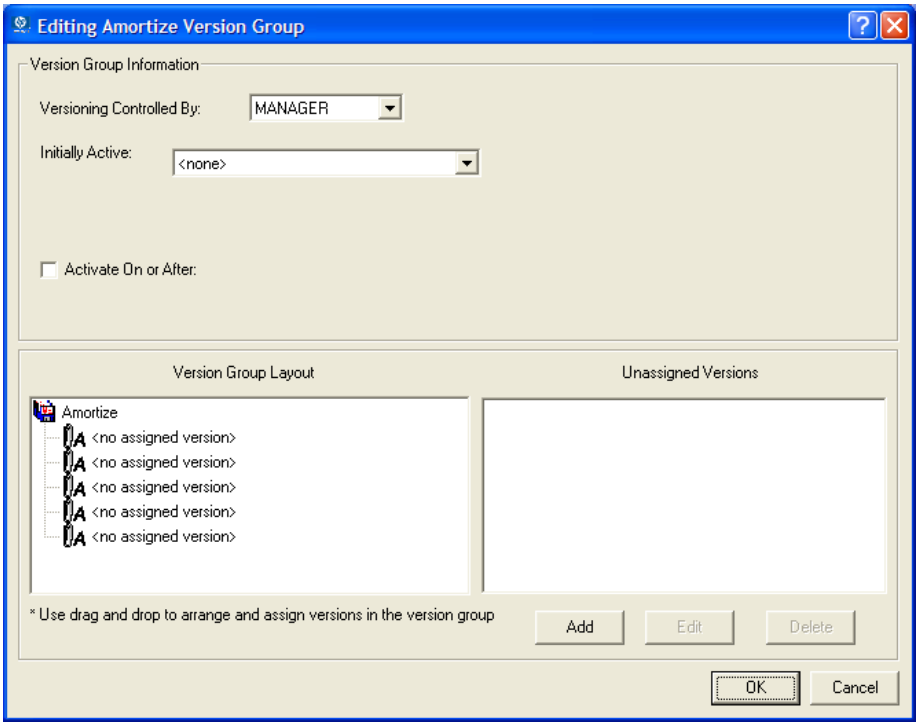
   The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This might have changed during installation. You can also change this by selecting the **Change Password** check box and typing the new password in the New Password and Verify New Password text boxes.

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

   The Admin CSDB Editor 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 window opens.

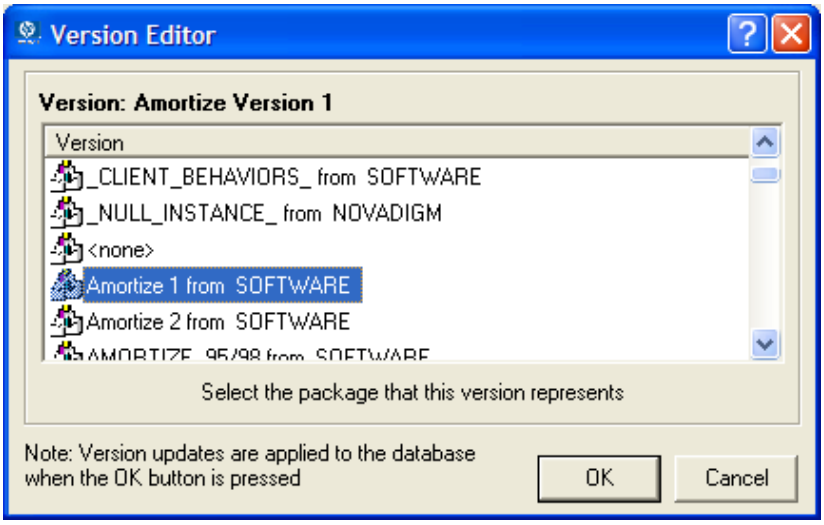


Creating a Version Instance

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

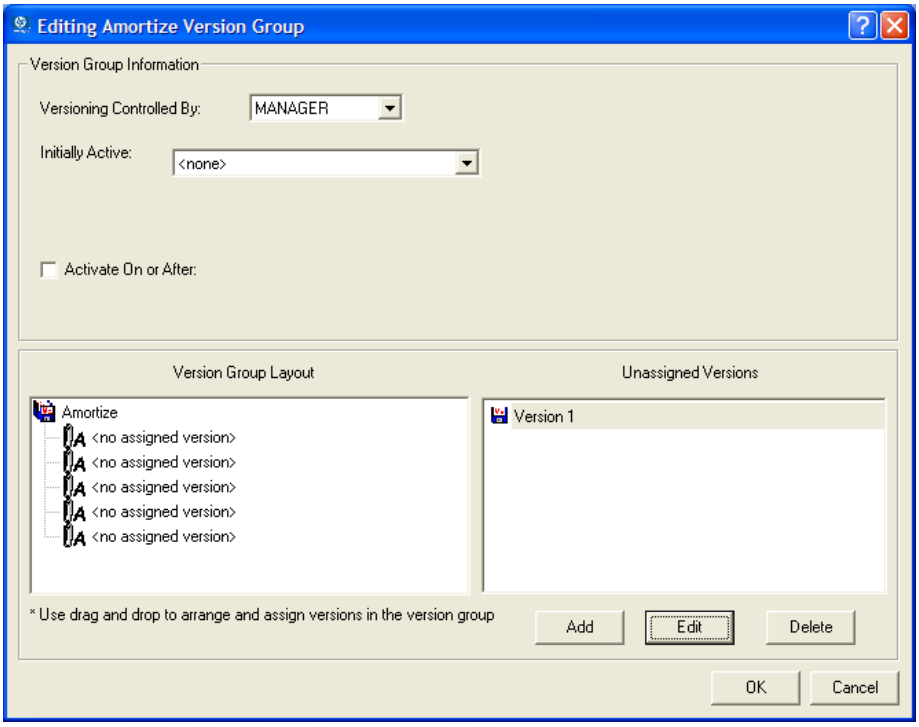
1. In the Version Group Editor, click **Add**. The Create Version dialog box opens.
2. Type a suffix that identifies the version. For example, type **`Version 1`**. The Version instance will be named Amortize Version 1.
3. Click **OK**. The Version Editor dialog box opens.



The Version Editor dialog box contains a list of Application Package (PACKAGE) instances stored in the Configuration Server DB. 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**.

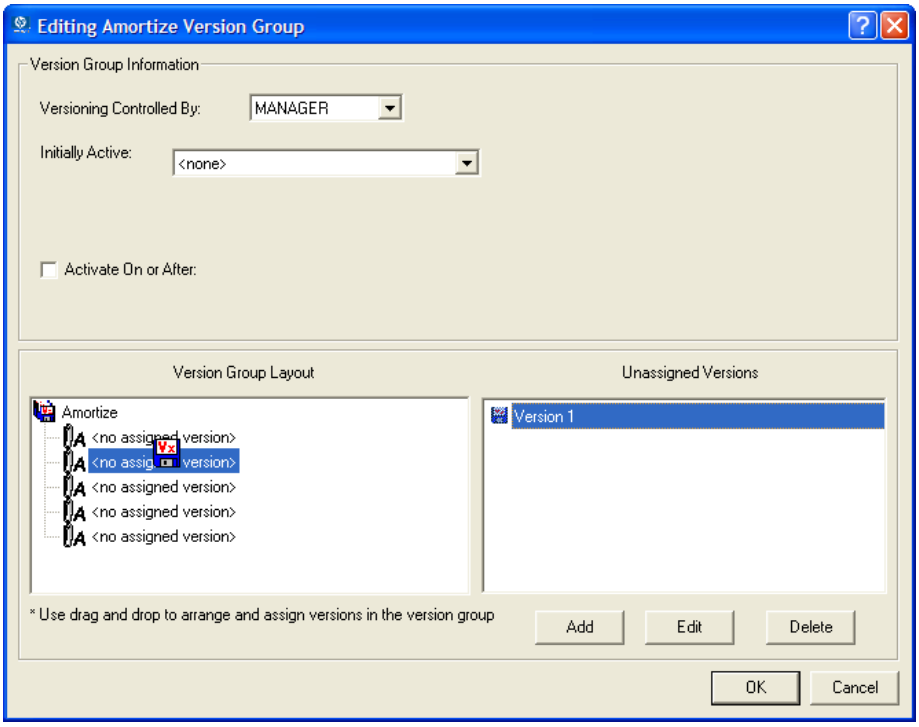
   The version instance will not be deleted until you close the Version Group Editor.
3. Click **OK** to close the Version Group Editor.

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**.

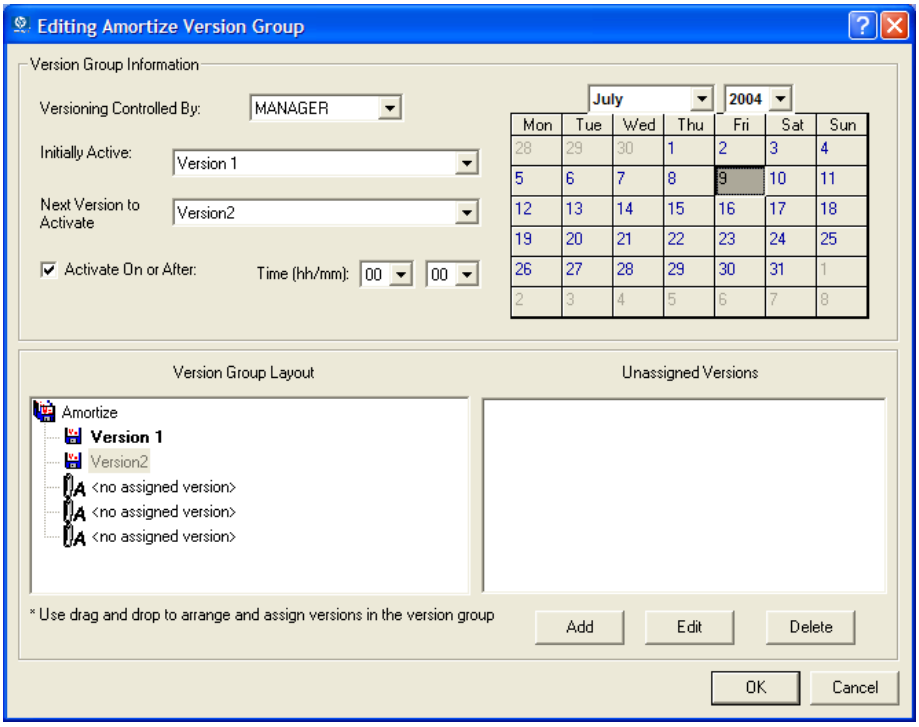
- Select **Manager** if you want to control the version to be deployed.
- Select **Client** if you want the subscriber to control the version to be deployed. *This is used only with the Application Self-service Manager agent.*

> 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.

2 In the Initially Active drop-down list, select the version that you want to activate on the agent device 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 select **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, HPCA 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

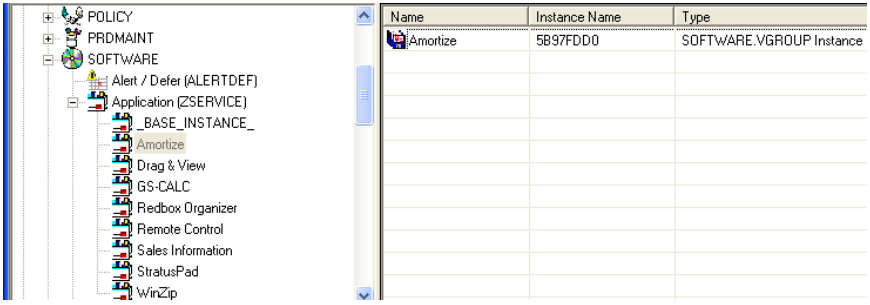
1 Navigate to PRIMARY.SOFTWARE.ZSERVICE.

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 Admin CSDB Editor.



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 agent device 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 agent device.

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.

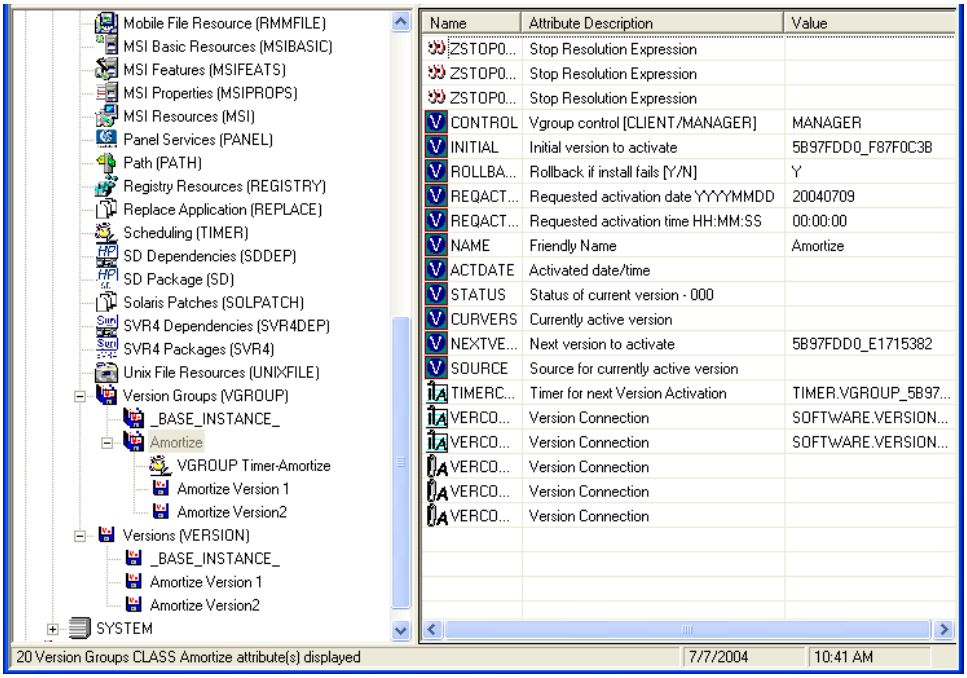
   or

   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 Admin CSDB Editor.

Figure 6 Version Groups (VGROUP) Class Instance



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

Table 28 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 Admin CSDB Editor to set this attribute.
CONTROL	Indicates whether the HPCA administrator (MANAGER) or the subscriber (CLIENT) controls which version to activate on the agent device. Use the Versioning Controlled By drop-down list in the Version Group Editor to set this option. Note: The Application Manager supports HPCA administrator-controlled version activation, but does not support subscriber-controlled activation.

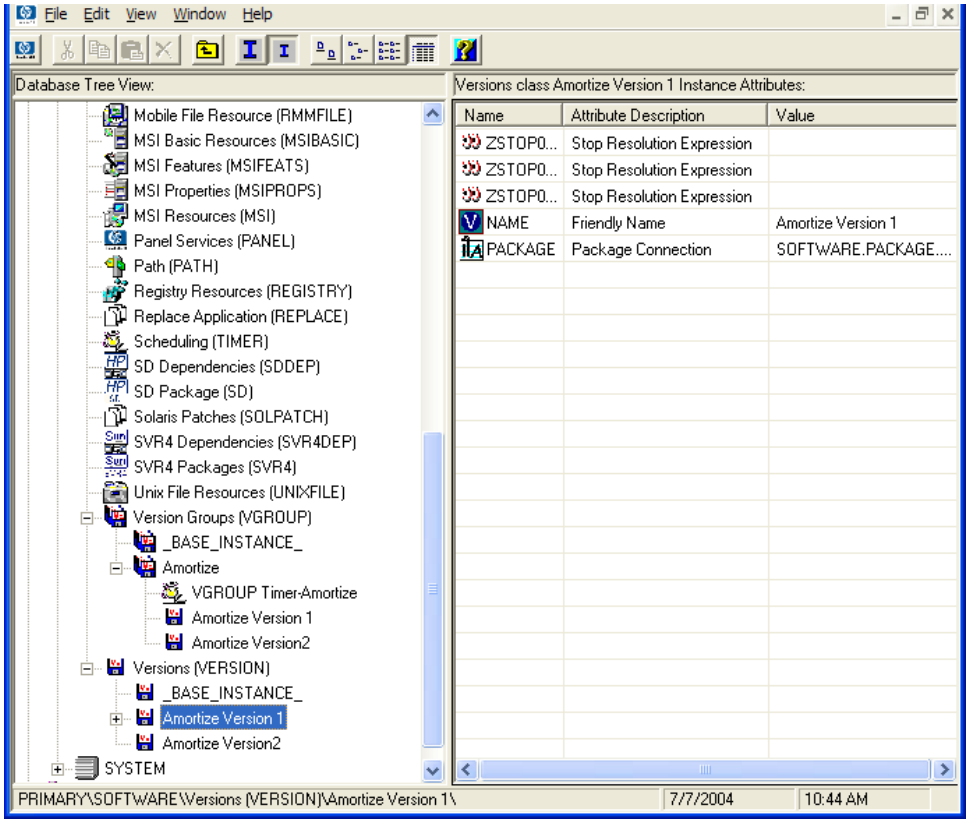
Attribute	Description
INITIAL	Indicates which version to activate on the agent device. Use the Initially Active drop-down list in the Version Group Editor to set this option.
ROLLBACK	Indicates whether to automatically roll back to the previously activated version when deployment of a new version fails. A new version may fail to deploy because of lack of sufficient disk space on the agent device, 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 agent device. If this attribute is blank, the version identified by the INITIAL attribute will be activated at the end of the agent connect that causes the version to be transferred to the agent device. 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 agent device. The version identified by the INITIAL attribute will be activated during the next agent 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 Admin CSDB Editor.
ACTDATE	Data maintained by the agent in the VGROUP object on the agent device. Do not enter a value. This is set by the agent.
STATUS	Data maintained by the agent in the VGROUP object on the agent device. Do not enter a value. This is set by the agent.
CURVERS	Data maintained by the agent in the VGROUP object on the agent device. Do not enter a value. This is set by the agent.
NEXTVERS	Data maintained by the agent in the VGROUP object on the agent device. Do not enter a value. This is set by the agent.

Attribute	Description
SOURCE	Data maintained by the agent in the VGROUP object on the agent device. Do not enter a value. This is set by the agent.
TIMERCON	If you specify a "next version to activate," the Admin CSDB Editor 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 HPCA. Use the Version Group Editor to create Versions class instances and assign them to a Version Group.

Figure 7 Versions (VERSION) Class instance



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

Table 29 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 Admin CSDB Editor to set this attribute.
NAME	The friendly name for the VERSION instance. This is set when you create the instance using the Version Group Editor.

Attribute	Description
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 Admin CSDB Editor. Many of the values for these attributes are set when using the Administrator, such as the Admin Publisher or the New Application Wizard in the Admin CSDB Editor. You can also use the Admin CSDB Editor 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 Admin CSDB Editor. The agent uses these attributes. For example, an attribute such as INSTDATE is used to record the date the service was installed on the agent device. The value for this attribute is stored in the PROFILE File for the agent device in the CSDB.

Table 30 Modifiable SOFTWARE.ZSERVICE attributes

Attribute	Description
ZSTOPnnn	**Expression Resolution Method** Stops resolution if the expression evaluates to TRUE. **Example**: WORDPOS(EDMGETV(ZMASTER,ZOS),'WINXP WIN2K NT')=0 This example expression will stop resolution on the instance if the agent device's operating system if the operating system is *not* Windows XP, Windows 2000, or Windows NT. In other words, the application will not be installed unless the agent device is running Windows XP, Windows 2000 or Windows NT.
ZSVCNAME	**Service Name/Description** Name of the service used for display in the Application Self-service Manager user interface. Value is set initially in the Short Description field in the New Application Wizard.

Attribute	Description
ZSVCTTYP	**Application Target Type [A/S]** Indicates which agent this application was packaged for, Application Manager or Application Self-service Manager. Value is set initially in the New Application Wizard. Possible values are A for Application Manager and S for Application Self-service 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 Application Self-service Manager, services are usually marked as optional, O. If you are using Application Manager *and* Application Self-service 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 Application Self-service 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. Refer to the *Administrator Guide* for more information on editing a class template. To process mandatory applications using Application Self-service 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.

Attribute	Description
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.*
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 Application Self-service Manager user interface. Value is set initially in the Author field in the New Application Wizard.

Attribute	Description
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 Application Self-service 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 Application Self-service 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 Application Self-service Manager user interface.
PRICE	**Price** Type in the price of an application to be displayed to subscribers in the extended information area in the Application Self-service Manager user interface.
SCHEDOK	**Update Schedule Locally [Y/N]** *For Application Self-service Manager only.* Specify **`Y`** to allow the subscriber to change the update schedule locally. Specify **`N`** to maintain control on the Configuration Server.
VERSION	**Version Description** Version of the software. This appears in the properties for the service in Application Self-service Manager user interface. The value is set initially in the Version field in the New Application Wizard.

Attribute	Description
NAME	**Friendly Name** This name appears in the properties for the service in the Application Self-service 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 agent device 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

Attribute	Description
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.
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 HPCA. • Set REMOVAL to D (Delete) to delete the service's objects and components. The service will still be managed by HPCA. • Set REMOVAL to U (Unmanage) to stop management of the service by HPCA. 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.

Attribute	Description
ZSVCCAT	**Service Visible in Catalog [Y/N]** Specifies whether the service is visible in the Application Self-service Manager Catalog. For optional applications, the default is Y. For mandatory applications, the default is N. Specify Y for Yes and N for No if you want to override these defaults.
UIOPTION	**Progress Indicator [NONE/FULL/INFO]** Controls whether the service status window appears. Possible values are: **NONE** = No interface appears. **FULL** = Interface appears and Cancel button is available. **INFO** = Interface appears with no option to cancel.
CACHE	**App Element Caching [Y/N]** Enables element caching. Specify Y for Yes and N for No. Default: N
CACHELOC	**CACHE Location On Client** *For Windows Installer applications only.* Location of the folder on the agent device that is used to cache the compressed application files needed for the product. HPCA support for Windows Installer tags the PRODGUID value to this value to create the folder. For example, If `CACHELOC=C:\progra~1\HP` and `PRODGUID = 12345_XXXX`, then the cache folder would be: `c:\progra~1\HP\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_

Attribute	Description
CACHELIM	**Percnt Disk Limit For Cache** *For Windows Installer applications only.* Cache limit, which is defined as the percentage of used drive space. Type a number between 000 and 100. If the percentage of used space is greater than the cache limit, then all of the cached files for the product are removed and the cache folder is deleted. This is checked after every file is cached on the disk.
ZDISCONN	**Disconnect on Install [Y/N]** Allows the agent 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).
ZSVCMODE	**Application Context [M/U/MU/EMU]** • Set ZSVCMODE to M if the service has only machine

Attribute	Description
	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 agent 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 Application Self-service 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 agent device's service objects. These attributes should *not* be modified using Admin CSDB Editor.

Table 31 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 Application Self-service 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 Application Self-service 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 agent 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 agent device. The agent manages and maintains this attribute. This is displayed to the subscribers in the extended information area in the Application Self-service 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 agent manages and maintains this attribute. It indicates when the application was last updated (in local time) on the agent device. 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 agent device. The agent manages and maintains this attribute. This is displayed to the subscribers in the extended information area in the Application Self-service 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 agent device. The agent manages and maintains this attribute. This attribute is useful for reporting purposes. The value is in the format of MMM DD,YYYY HH:MM:SS. Example: Jul 28, 2003 16:10:00

Summary

- Set the appropriate context, M, U, MU, or EMU, for a service using the ZSVCMODE attribute.
- If a service requires a reboot of the agent device, use the REBOOT attribute in the Application (ZSERVICE) Class.
- Be aware of all of your service options in the Application (ZSERVICE) Class.

7 Deploying Services

At the end of this chapter, you will:

- Understand the different deployment methods available in HPCA 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 describes the *suggested* implementation for the HP Client Automation Agents; Application Manager and Application Self-service 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 these agents. This chapter describes deploying services.

About Deployment Methods

After creating a service 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 HPCA, 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 HPCA.

- **Scheduler**
  Installs the service at a specific time or sets any command line to run at an interval.
- **Notify**
  Forces one or more agent devices 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.

In addition to the methods listed above, applications can be deployed using the Application Self-Service Manager user-interface. This method of deploying optional applications is described in the next chapter, HPCA Application Self-service Manager User Interface.

Before selecting a deployment method, consider the following.

- Does the application need to be deployed at a certain time? If so, use Scheduler.
- Do you want to notify the users via e-mail when you are deploying the application? If so, use Notify.
- Do you want to install a new application, an update to an application, or remove an application? If so, use Notify.
- Are there multiple versions of the application? If so, use Version Groups. See Chapter 6, Preparing Services.

Testing Deployments

To ensure successful deployments, test your implementation rigorously.

- 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. AM stands for Application Manager and ASM stands for Application Self-service 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 agent device for policy.

Table 32 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. AM default: prompt. ASM default: Depends on request type.

Parameter	Explanation
dname	The Configuration Server Database (CSDB) domain name for the services. This is the directory under which the service catalog (`ASERVICE.EDM`) is stored. For example, dname=SOFTWARE. Software AM default: SOFTWARE. If preload=y, then the default is RADSTAGE. ASM 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, HPCA uses the IP address specified in the ZMASTER object stored in IDMLIB (by default, /opt/HP/CM/Agent/lib). AM default: NOVARCS (only defaults if no arguments are passed). ASM default: NOVARCS (only defaults if *no* arguments are passed).
mname	Name of the Configuration Server. For example, mname=RADSVR01. AM default: CM (defaults to RADSTAGE for preload). ASM default: CM (defaults to RADSTAGE for preload).
port	Configuration Server port. The default for this is 3464. Note: If you do not specify the port, HPCA uses the port specified in the ZMASTER object stored in IDMLIB (by default, /opt/HP/CM/lib). AM default: 3464. ASM 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. AM default: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). ASM default: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). Application Self-service Manager does not pass a context by default.
uid	Identification used to identify the current session. Note: If uid is set on the command line, and startdir is not, then the startdir will be set to the same value as uid. If you specify a uid on the command line, we recommend specifying the startdir as well. • `uid = $MACHINE` identifies the current session by the name of the computer. • `uid = $USER` identifies the current session by the name of the user currently logged on. • `uid=custom` is used to identify the current session by a custom value that you specify. AM default: $USER if started in a user context (context=u). SYSTEM if started in machine context (context=m). ASM 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, HPCA uses the LOCALUID specified in the ZMASTER object stored in IDMLIB (by default, `/opt/HP/CM/Agent/lib`).

Operations

These parameters influence how client will connect. Features include computer restart handling, log specifications, and the display options for the subscriber.

Table 33 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. AM default: Y if System Tray is running. N if System Tray is *not* running or there are no users logged on. ASM 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 agent device will shut down no matter what the reboot settings are for a particular service. Note: This replaces handle_reboot. AM default: Y ASM 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. AM default: Y ASM 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. AM default: UserConnect if started in a USER context. MachineConnect if started in SYSTEM context. ASM 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. AM default: 1000000 bytes ASM default: 1000000 bytes
rtimeout	Specify number of seconds to wait if a reboot panel has been requested for a service before rebooting the agent device. 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 agent device. 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 34 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`. AM default: prompt ASM default: Depends on request type

Parameter	Explanation

context	Set `context = m` when installing an application in the machine context.
	If `context = m` then the following defaults are assumed:
	`uid=$machine`
	`startdir=system`
	`cat=prompt`
	`ulogon=n`
	Set `context = u` when installing an application in the user context.
	If `context = u` then the following defaults are assumed:
	`startdir=$user`
	`uid=$user`
	`cat=prompt`
	`ulogon=y`
	AM default: If started with a user logged on, the context defaults to u. If no user is logged on, then context defaults to m.
	ASM default: There is no default for Application Self-service Manager. All components are processed.

flushu	If you are using `local=y`, set `flushu=y` on user connects (`context=u`) to send reporting objects up to the 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. Note: `flushu=n` is not applicable for machine connect. AM default: Y ASM default: Y

Parameter	Explanation
local	Set this to y to install resources for the user's services from the local agent device. Use this only with `context = u`. Usually, this is used with `cat = m`.
machfreq	(4.0 Feature) Use this variable to prevent HPCA from running every time an agent device 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 agent will not run more than once within the specified timeframe to reduce the number of ROM commits on a thin agent device. If you set MACHFREQ to 0, the machine connect will run on *every* reboot of a thin client.
ulogon	*Only used if cat = prompt.* Set `ulogon = n` to hide the logon panel. Note: If using System Tray, set ulogon = n. This will display the HPCA logon panel, which is not supported by System Tray. AM default: N ASM default: Y
userfreq	*Only used if context=u.* Use this variable to prevent HPCA from running every time a user logs into the agent device. 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 an agent connect is run with `context = u`.

Client Operations Profiles

These parameters are used for specifying how to use Client Operations Profiles. For more information, see Chapter 5, Configuring Client Operations Profiles.

Table 35 radskman Client Operations Profiles Parameters

Parameter	Explanation
cop	(4.0 Feature) Set to Y to enable Client Operations Profile resolution for this agent connect only. Set to N to disable Client Operations Profiles resolution for this agent 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. AM default: N ASM default: N
datauri	(4.0 Feature) If you want to override the use of the SAP object for the Data Type, add `datauri` to the radskman command line. Datauri should be in the same format as the Universal Resource Identifier. For the syntax of this parameter, see Table 20 on page 94.
product	(4.0 Feature) If you used the SAP.PRODUCT attribute to identify that a SAP can only be used with a specific product, specify that product using this parameter. For example, if SAP.PRODUCT is set to IM, set `product=RIM` on the radskman command line. Specify multiple product filters separated by a comma.
rcsuri	(4.0 Feature) If you want to override the use of the SAP object for the RCS Type, add `rcsuri` to the radskman command line. `Rcsuri` should be in the same format as the Universal Resource Identifier. For the syntax of this parameter, see Table 20 on page 94.

Process

Process parameters involve service processing such as whether to repair or add applications on the current agent 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 agents.

Table 36 radskman Process Parameters

Parameter	Explanation
Add	Set this to N if you do not want to install applications during this agent connect. AM default: Y ASM default: Y
autofix	Set `autofix = y` to automatically repair any broken applications. Set `autofix = n` to prevent broken applications from being fixed. AM default: Y ASM default: Y
catexp	(4.0 Feature) Use this parameter to process applications based on a particular attribute in the ZSERVICE Class. Use the format *`attribute name`*`:`*`value`*. Specify multiple OR conditions with a forward slash (`/`). For example, to process only applications that have the ZSERVICE.CATGROUP attribute set to finance, set catexp=catgroup:finance.
del	Set this to N if you do not want to delete applications during this agent connect. AM default: Y ASM 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 Agent Self Maintenance on this connect. Note: Agent Self Maintenance will not be deployed unless you set `mnt=Y`. AM default: N ASM default: N
preload	Use this for Proxy 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`.

Parameter	Explanation
rep	Set this to N if you do not want to repair applications during this agent connect session. AM default: Y ASM default: Y
sendcat	Set this to **y** to send the service list, stored in the agent device's ASERVICE object, to the Configuration Server at the end of the agent 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 agent 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 agent connect session. AM default: Y ASM default: Y
ver	Set this to N if you do not want to verify applications during this agent connect session. AM default: Y ASM default: Y

radskman Examples:

The following examples are provided to illustrate common uses of radskman.

```
radskman ip=10.10.10.15,port=3464,mname=hpca-cs,dname=software,
cat=prompt
```

Performs a first catalog refresh that brings down the catalog (`aservice.edm`), runs self-maintenance, does not display the user logon panel if using Application Manager, and processes all mandatory applications:

```
radskman ip=test.corp.com,port=3464,mname=hpca-cs,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=hpca-cs,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=hpca-cs,dname=software,
cat=y,sname=WINZIP
```

Install a single application with the service name of WINZIP, while only updating the catalog. Note: In this example, the Configuration Server uses a custom port number.

```
radskman ip=10.10.10.15,port=3464,mname=hpca-cs,dname=software,
cat=prompt,hreboot=Y,ask=Y
```

Process all mandatory applications, handle reboot requests, and prompt the user with a panel to confirm the reboot request:

```
radskman ip=10.10.10.15,port=3464,uid=STAGER,preload=Y
```

Silently preload a Staging Server using the default location of RADSTAGE. This syntax is often run by a daily or weekly timer instance that is deployed to an agent that is co-located on the same machine as the Staging Server. Note: The preload parameter automatically suppresses the user logon panel and progress indicator panels. The preload parameter also does not run any methods (ZCREATE, ZVERIFY, etc.) or evaluate any expressions on the Configuration Server.

```
radskman ip=10.10.10.15,port=3464,uid=STAGER,preload=d:
\stager,ind=Y
```

Preload a stager using a location of `d:\stager` and display the progress indicator panels. Note: If the System Tray feature is enabled, then the progress indicator will be displayed in the System Tray information bubble. If the System Tray is disabled, then the progress indicator will be displayed in a separate panel.

```
radskman context=m
```

Perform a machine connect. Because context was specified as m, and no other parameters were passed the following default values are used: `ip=NOVARCS`, `port=3464`, `uid=$machine`, `startdir=system`, `cat=prompt`, `ulogon=n`, `mname=hpca-cs`, `dname=software`.

`radskman context=u`

Perform a user connect. Because context was specified as u, and no other parameters were passed the following default values are used: `ip=NOVARCS`, `port=3464`, `uid=$user`, `startdir=$user`, `cat=prompt`, `ulogon=y`, `mname=hpca-cs`, `dname=software`.

`radskman context=u,userfreq=12`

Perform a user connect only if a machine connect has occurred since the last user connect and there has been at least 12 hours since the last user connect.

Deployment Methods

The following section covers each of the deployment methods in detail. Select the appropriate method for your subscribers. Remember, you can use multiple deployment methods to distribute a single application.

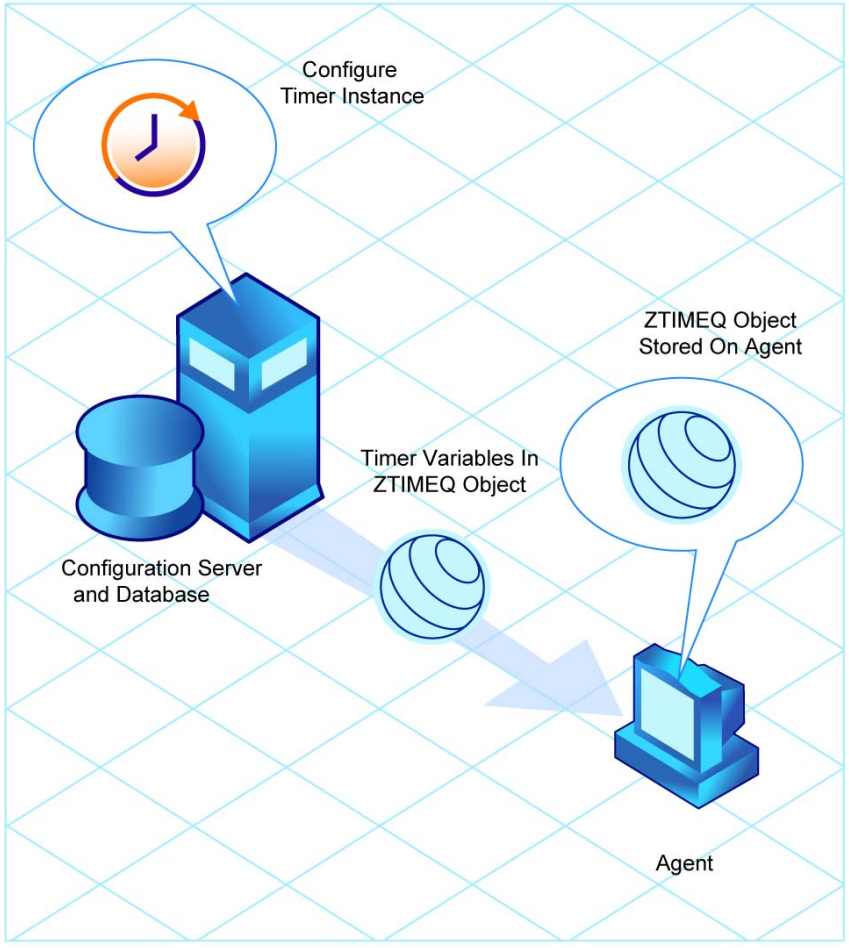
Scheduling (TIMER)

The Scheduler service, **radsched**, is installed with the Application Manager. The 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 Linux workstations. We recommend running the radsched daemon as root. Consult your company's 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. The next time the agent device connects to the Configuration Server, the ZTIMEQ object is created on the agent device, and the timer variables are transferred to the ZTIMEQ object.

Figure 8 Transferring the timer instance



This section describes 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 agent 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 agent device 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 Admin CSDB Editor to create a Scheduling (TIMER) instance in the SOFTWARE Domain.

To create a new timer in the SOFTWARE Domain

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**.

The Admin CSDB Editor Security Information dialog box opens.

The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This might have been changed during installation. 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.

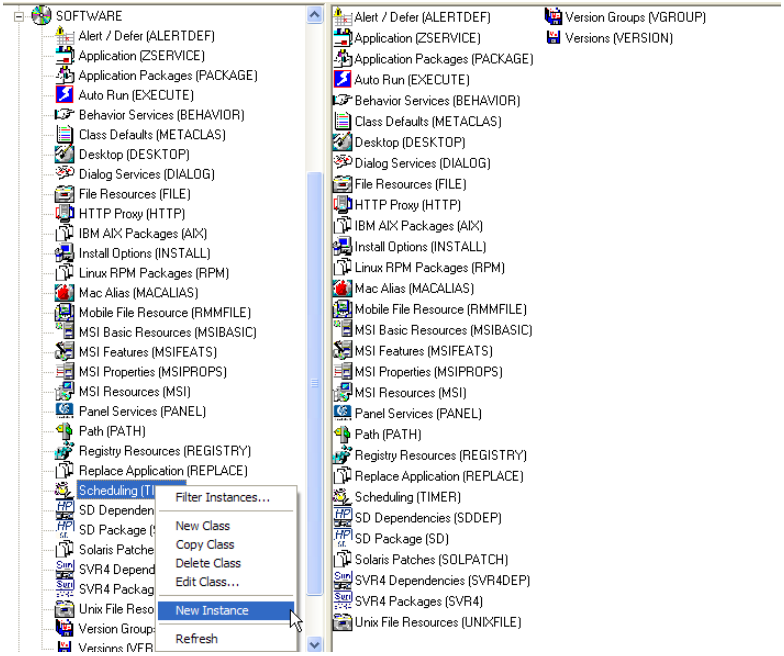
2 If necessary, type a User ID and Password, and then click **OK**.

3 The Admin CSDB Editor 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 agent device. The following table describes these attributes.

Table 37 Scheduling (TIMER) attributes

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

Attribute	Usage
NETAVAIL	If set to Y, check for network availability before executing the TIMER instance. If the network is not available, network availability will be checked every time the timer wakes up until the network is available. If set to N, the TIMER instance will be executed without checking for network availability. If set to W, check for network availability before executing the TIMER instance. If the network is not available and the time window's end limit has been reached, the timer will wait until the next time window before checking for network availability again. The default is N.
PINGDLAY	If ZNOPING is set to N, PINGDLAY specifies the time in milliseconds between pings. The default is 2000.
PINGCNT	If ZNOPING is set to N, PINGCNT specifies number of ping attempts. The default is 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 agent device 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 agent device when the timer expires. Use radskman to verify and update HPCA-managed mandatory applications. See Connection Parameters (radskman) on page 159 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 the Timer Expiration on page 180 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 185 for more information.
ZSCHTYPE	This attribute is valid only when **`ZSCHFREQ=PERIODIC`**. Valid values are IMMEDIATE and DEFERRED. Specify **`DEFERRED`** to indicate that the first time an event is attempted to be launched, it will be deferred until the next scheduled time, regardless of when the timer instance is evaluated. This was designed so that events that are scheduled for off-peak hours will not launch while a user is working. **Example 1** Assume a timer with **`ZSCHDEF=DAILY(&ZSYSDATE,04:00:00)`**. If **`ZSCHTYPE=IMMEDIATE`** and it is: — Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00

Attribute	Usage
	a.m. — Later than 4:00 a.m., the command in the instance will be executed immediately. If **ZSCHTYPE=DEFERRED** and it is: — Earlier than 4:00 a.m., the command in the instance will be executed the same day at 4:00 a.m. — Later than 4:00 a.m., the command in the instance will be executed the next day at 4:00 a.m. **Example 2** Assume a timer with **ZSCHDEF=WEEKDAY(FRIDAY,04:00:00)**. If **ZSCHTYPE=IMMEDIATE** and it is: — Either not Friday, or earlier than 4:00 a.m. on Friday, the command in the instance will be executed on Friday at 4:00 a.m. — Later than 4:00 a.m. on Friday, the command in the instance will be executed immediately. If **ZSCHTYPE=DEFERRED** and it is: — Not Friday, the command in the instance will be executed on the next occurring Friday, at 4:00 a.m. — Earlier than 4:00 a.m. on Friday, the command in the instance will be deferred one week and executed a week later on the following Friday, at 4:00 a.m. — Later than 4:00 a.m. on Friday, the command in the instance will be executed a week later on Friday at 4:00 a.m.
ZSTOP	Expressions evaluating to "true" in ZSTOP attributes cause resolution of the instance to be skipped. If left blank, the instance is accepted, and resolution continues. This is useful if you want to set conditions on which of your subscribers receive the timer.
The values for the following attributes are set from the BASE INSTANCE of the TIMER class and should not be edited.	

Attribute	Usage
RUNSYNC	Specifies if synchronous timer execution will take place. The default value is Y.
ZOBJPRI	Indicates the priority for deployment of the ZTIMEQ object, relative to the other elements deployed during the agent connect. Elements with priority numbers lower than the value of ZOBJPRI are deployed before this ZTIMEQ object. A value of 90 is inherited from the BASE INSTANCE.
ZSCHMODE	Specifies the timer owner. Leave as Default.
ZSVCOID	Specifies the object ID of the Application instance that this Scheduling instance is connected to. The value is inherited from the BASE INSTANCE.
ZCHNNAME	Specifies the name of the domain in the CSDB 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 agent device. The value is inherited from the BASE INSTANCE.
ZVERIFY	The Scheduler Verify method that runs on the agent device. The value is inherited from the BASE INSTANCE.
ZUPDATE	The Scheduler Update method that runs on the agent device. The value is inherited from the BASE INSTANCE.
ZDELETE	The Scheduler Delete method that runs on the agent device. The value is inherited from the BASE INSTANCE.

Configuring the Timer

This section offers a review of the syntax that is used to configure the attributes of the TIMER instance. Following that, in the section, Deploying Applications over a Period of Time, is a sample exercise on how to configure a TIMER instance to deploy mandatory applications during off-peak hours.

Specifying the Timer Expiration (ZSCHDEF)

Use the ZSCHDEF and ZSCHFREQ attributes to specify when and how often a timer will expire.

- ZSCHDEF indicates when the timer will expire;
- ZSCHFREQ indicates how often the timer will expire.

The syntax of the ZSCHFREQ attribute will influence the settings of the ZSCHDEF attribute. Use Table 38 on page 181 to determine the appropriate syntax for the value of ZSCHDEF. Before configuring the ZSCHDEF attribute, review the following syntax-formatting considerations.

- The value of WEEKDAY must be UPPERCASED and will accept only the days of the week: MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY, and SUNDAY.
- In all ZSCHDEF attribute syntax, the *time* value must be expressed in base-24 time, in the format **HH:MM:SS**. Valid values are **00:00:00**–**23:59:59**.
- In all ZSCHDEF attribute syntax, the *date* value must be expressed in the format **YYYYMMDD**.

Limit Time Parameter

The *limit time* parameter is used to specify a time after which the command will not be executed. For example, the command

```
DAILY(20070707,18:00:00[,20:00:00])
```

will execute any time between 6 p.m. and 8 p.m. on July 7, 2007, but it will not execute after 8 p.m. So, if the target machine is not powered on (or the Scheduler is not running) during this time, the command will not execute. It will be rescheduled for the next occurrence of "between 6 p.m. and 8 p.m." which, in this case, because it is a "daily" command, will execute on the next day, July 8, 2007.

ZSCHFREQ=RANDOM

When **ZSCHFREQ=RANDOM** is specified the *time* parameter is automatically replaced with two parameters, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*.

Be sure to not use the *limit time* parameter as an *end time* indicator; their functionalities are different and doing so will result in a malformed ZSCHDEF command.

If `ZSCHFREQ=RANDOM` and the *`limit time`* parameter is NOT specified: The *`end time`* parameter can span midnight (it can be the next day). For example, the commands

`DAILY(20070707,20:00:00,06:00:00)` and

`NUMDAYS(20070707,20:00:00,06:00:00,,14)`

will execute at random times between 8 p.m. on July 7, 2007 and 6 a.m. on July 8, 2007. Note that even though the *`limit time`* parameter is not specified in either command, NUMDAYS still requires the third comma in order to be considered a valid argument; DAILY does not.

If `ZSCHFREQ=RANDOM` and the *`limit time`* parameter IS specified: The *`end time`* parameter cannot span midnight. If it does, the RADTIMEQ create method will log a warning and set the *`start time`* to midnight (00:00:00).

Table 38 ZSCHDEF Attribute Syntax

Attribute	Description
HOURLY	The timer will expire and run hourly, starting any time after the specified time but not later that the specified limit time, based on the system's date. **Syntax**: `&SYSDATE,`*`time`*`[,`*`limit time`*`]` **Example**: **`ZSCHDEF=HOURLY(&ZSYSDATE,04:30:00)`** **Note**: If **`ZSCHFREQ=RANDOM`**, the *`start time`* and *`end time`* parameters are activated. **Syntax**: `&SYSDATE,`*`start time`*`,`*`end time`*`[,`*`limit time`*`]` **Example**: **`ZSCHDEF=HOURLY(&ZSYSDATE,04:30:00,09:00:00)`** For more information, see the section Specifying the Timer Expiration on page 180.
DAILY	The timer will expire and run daily at the specified time (but not later that the specified limit time), based on the system's date. **Syntax**: `&SYSDATE,`*`time`*`[,`*`limit time`*`]` **Example**: **`ZSCHDEF=DAILY(&ZSYSDATE,12:00:00)`**

Attribute	Description
	Note: If **ZSCHFREQ=RANDOM**, the *start time* and *end time* parameters are activated. **Syntax**: &SYSDATE,*start time*,*end time*[,*limit time*] **Example**: **ZSCHDEF=DAILY(&ZSYSDATE,12:00:00,14:00:00, 18:00:00)** For more information, see the section Specifying the Timer Expiration on page 180.
WEEKLY	The timer will expire and run at the specified time (but not later that the specified limit time) on every seventh day, based on the system's date. **Syntax**: &SYSDATE,*time*[,*limit time*] **Example**: **ZSCHDEF=WEEKLY(&ZSYSDATE,08:00:00)** **Note**: If **ZSCHFREQ=RANDOM**, the *start time* and *end time* parameters are activated. **Syntax**: &SYSDATE,*start time*,*end time*[,*limit time*] **Example**: **ZSCHDEF=WEEKLY(&ZSYSDATE,08:00:00,12:00:00, 14:00:00)** For more information, see the section Specifying the Timer Expiration on page 180.
INTERVAL	The timer will expire and run every *n* minutes starting at the specified time (but not later that the specified limit time), based on the system's date. **Syntax**: &SYSDATE,*time*,[*limit time*],*interval* **Example**: **ZSCHDEF=INTERVAL(&ZSYSDATE,04:00:00,06:00:00,30)** **Notes**: INTERVAL must be specified in minutes. The third comma is required regardless of whether a third argument is specified. If **ZSCHFREQ=RANDOM**, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*. **Syntax**: &SYSDATE,*start time*,*end time*,[*limit time*],*interval* **Example**: **ZSCHDEF=INTERVAL(&ZSYSDATE,04:00:00,23:00:00, 06:00:00,30)** For more information, see the section Specifying the Timer Expiration on page 180.

Attribute	Description
WEEKDAY	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified weekday, every week. **Syntax**: *WEEKDAY*,*time*[,*limit time*] **Example**: **ZSCHDEF=WEEKDAY(TUESDAY,01:00:00)** **Note**: If **ZSCHFREQ=RANDOM**, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*. **Syntax**: *WEEKDAY*,*start time*,*end time*[,*limit time*] **Example**: **ZSCHDEF=INTERVAL(TUESDAY,04:00:00,06:00:00, 10:00:00)** For more information, see the section Specifying the Timer Expiration on page 180.
MONTHDAY	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified weekday in the week of the month that is indicated by the 4th parameter. **Syntax**: *WEEKDAY*,*time*,[*limit time*],*week of the month* **Example**: **ZSCHDEF=MONTHDAY(TUESDAY,01:00:00,,2)** **Notes**: The valid values for the 4th parameter are 1–5. If this argument is not specified, the timer will expire during the first week of the month. The third comma is required regardless of whether a third argument is specified. If **ZSCHFREQ=RANDOM**, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*. **Syntax**: *WEEKDAY*,*start time*,*end time*,[*limit time*],*week of the month* **Example**: **ZSCHDEF=MONTHDAY(TUESDAY,01:00:00,04:00:00,,2)** **Important Note**: Consider the consequences of specifying an *end time* that spans midnight (occurs on the following day). For more information, see the section, Specifying the Timer Expiration on page 180.
MONTHLY	The timer will expire and run at the specified time (but not later that the specified limit time) on the *n*th of every month, starting in the specified month and year. **Syntax**: *date*,*time*[,*limit time*] **Example**: **ZSCHDEF=MONTHLY(20040215,01:00:00,05:30:00)**

Attribute	Description
	Note: If **ZSCHFREQ=RANDOM**, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*. See the section, Specifying the Timer Expiration on page 180, for more information. **Syntax**: *date,start time,end time*[,*limit time*] **Example**: **ZSCHDEF=MONTHLY(20040215,01:00:00,05:30:00, 07:00:00)** **Important Note**: This attribute reschedules differently than other ZSCHDEF attributes; it will reschedule by adjusting the month (but retaining the date) for which it was originally scheduled, rather than adjusting the date based on when it eventually ran. For example, assume **ZSCHDEF=MONTHLY(20040116,05:30:00)** and that the agent device was powered off on January 16th and that the timer didn't execute until January 18th. The new schedule would automatically revise to **MONTHLY(20040216,05:30:00)** rather than **MONTHLY(20040218,05:30:00)**.
NUMDAYS	The timer will expire and run at the specified time (but not later that the specified limit time) on the specified date, then again on every *n*th day (as specified by the 4th parameter). **Syntax**: *date,time,*[*limit time*],*number of days* **Example**: **ZSCHDEF=NUMDAYS(20040803,18:00:00,21:30:00,14)** **Notes**: The third comma is required regardless of whether a third argument is specified. If **ZSCHFREQ=RANDOM**, the *time* parameter is automatically replaced with two parameters, *start time* and *end time*. **Syntax**: *date,start time,end time,*[*limit time*],*number of days* **Example**: **ZSCHDEF=NUMDAYS(20040803,18:00:00,21:30:00, 22:00:00,14)** For more information, see the section, Specifying the Timer Expiration on page 180.

Attribute	Description
STARTUP	When the Scheduler starts on the agent device, it will immediately execute all Timer instances that have ZSCHDEF=STARTUP specified. It will check for special conditions such as NETAVAIL, ZNOPING, and RETRYFLG. After executing all the STARTUP instances, RADSCHED will return to its regular timer loop. It will execute STARTUP instances in the regular timer loop only 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

Applications can be deployed over a period of time in order to balance the workload on the Configuration Server and alleviate network congestion.

To do this, configure the timer for "random" expiration and use ZSCHDEF to specify the period of time during which the applications should be deployed. The time-period options are detailed in Table 38 starting on page 181.

In the following example, a timer will be configured to deploy mandatory applications on a weekly basis. The deployments will be scheduled to run between 5:00 p.m. and 7:00 p.m. in order to alleviate network congestion.

To specify when the timer expires

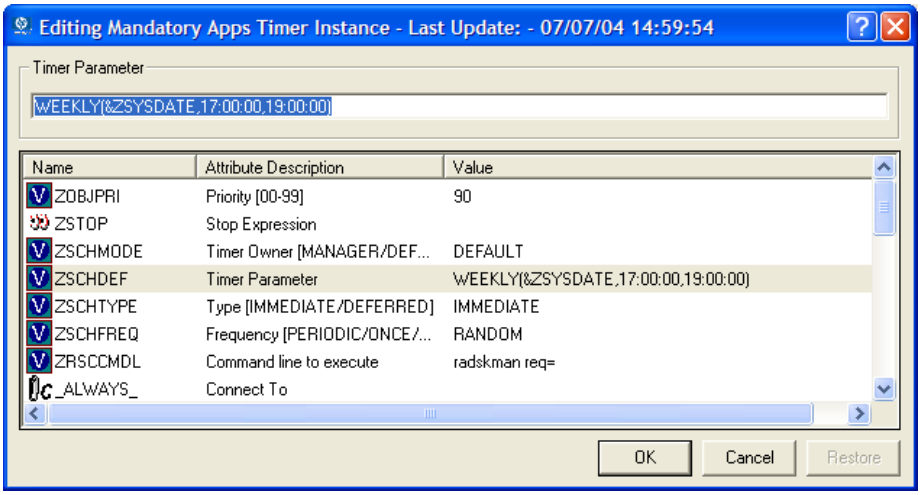
1. After navigating to the timer instance, double-click **ZSCHFREQ** in the list view of the Admin CSDB Editor.

   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.

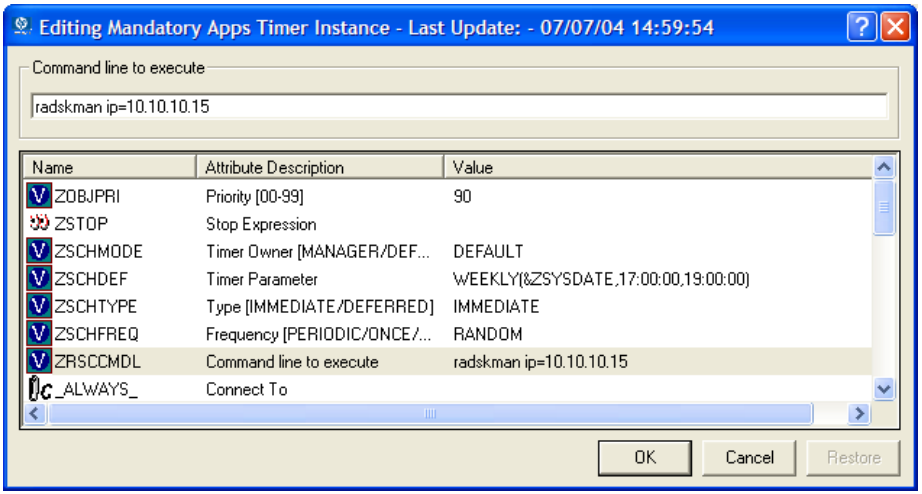
Specifying the Command Line (ZRSCCMDL)

When the timer expires, it executes any command line that you specify on the agent device.

To see how timers work, 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 agent device, 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 Admin CSDB Editor. 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 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 agent 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 Admin CSDB Editor, double-click **PRIMARY**.

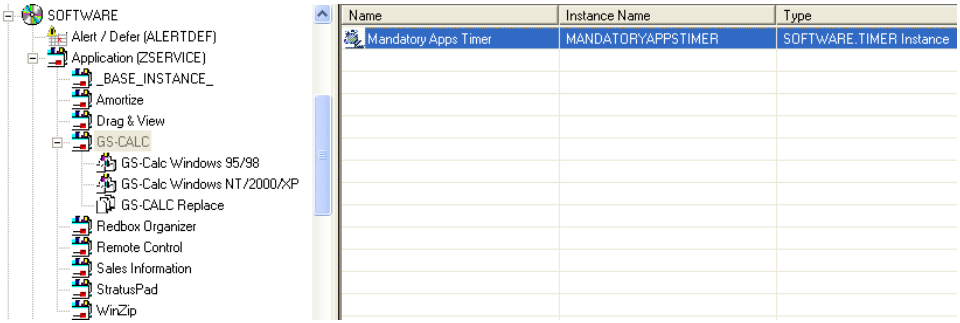
3 Double-click **SOFTWARE**.

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

5 Right-click on the Application class (in this example, 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-Calc service to the Mandatory Apps Timer.

11 Click **OK** to close the confirmation message.

Testing the Timer Deployment

The first time an agent device connects to the Configuration Server after the timer is created, the timer information is transferred to the agent device in the ZTIMEQ object. The ZTIMEQ object is located in the IDMROOT directory

In this section, we will force the agent device 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 agent device that updates *all* mandatory services and performs any agent self-maintenance.

To connect to the Configuration Server

1 On the agent device, go to a command prompt and change the directory to `/opt/HP/CM/Agent`. This is the default location for radskman.

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

See Connection Parameters (radskman) on page 159 for information about radskman and the parameters above.

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

3 Press **Enter**. Once the agent connect is finished, you can view the ZTIMEQ object on the agent device.

Viewing the Timer Object (ZTIMEQ.EDM)

Now that we have forced the agent device to connect to the Configuration Server, the ZTIMEQ object is stored on agent device. Use the Admin Agent Explorer, installed as part of the Administrator, to view or modify the ZTIMEQ object.

After the timer expires, the ZTIMEQ object is removed from the agent device during the next agent 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 agent 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 agent connect.

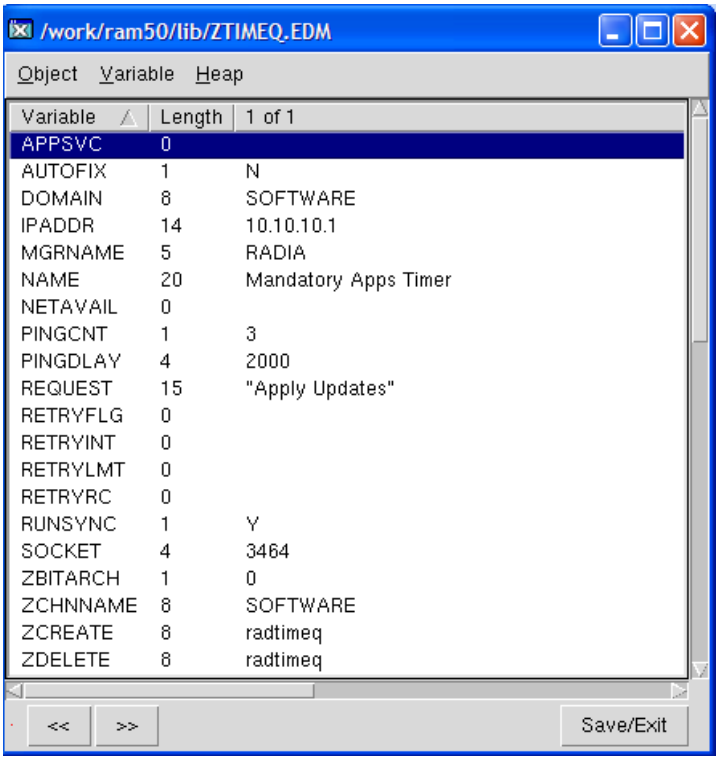
The ZTIMEQ object contains one instance for each Scheduling (TIMER) instance in the Configuration Server Database (CSDB). For example, if two different services each have timer instances associated with them, then there will be two instances in the ZTIMEQ object.

To view the ZTIMEQ object on the agent device

1 Go to the directory where you installed the Administrator.

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

3 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 agent device to see what happens in various situations.

To edit an attribute in ZTIMEQ

1 Double-click the attribute that you want to edit.

The Change Variable dialog box opens.

2 Type the new value.

3 Click **Save/Exit**.

To quickly determine whether the timer expires, you can change ZRSCCMDL to run any executable, such as Notepad. When the timer expires, Notepad opens, confirming that the timer expired.

Timer Logs

Timer events are tracked in three logs, stored in the `IDMLOG` directory (by default, `/opt/HP/CM/Agent/log`).

The following table describes the timer logs.

Table 39 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 agent 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 Notify to force one or more agent devices to connect to the Configuration Server to install, update, or remove an application. Each agent device runs the Notify service in the background. This service waits to receive a Notify message from the Configuration Server. When a message is received, the agent device connects to the Configuration Server and performs the action initiated by the Notify operation. Notify can also send e-mail notification to agent devices.

You can initiate a Notify by:

- Selecting **Notify Subscribers** from the shortcut menu for an Application (ZSERVICE) instance. Use this option t *only* 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 agent device to perform the Notify.

  Drag-and-Drop Notify is intended for use in environments with a single Configuration Server.

Requirements for Using Notify

To use Notify

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

  Notify is designed to notify only subscribers whose information is in the PROFILE File in the CSDB.

- 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/HP/CM/Agent`).

- 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

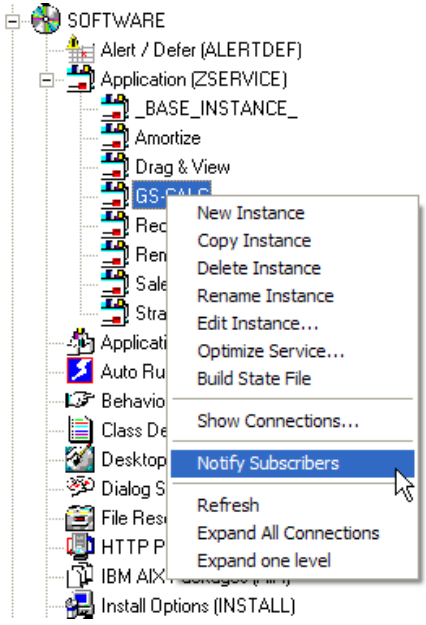
To update or remove a service, initiate the Notify from the shortcut menu for the Application (ZSERVICE) instance.

Notify communicates with agent devices that are members of an **audience list**. An agent device is added to the audience list when HPCA 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.



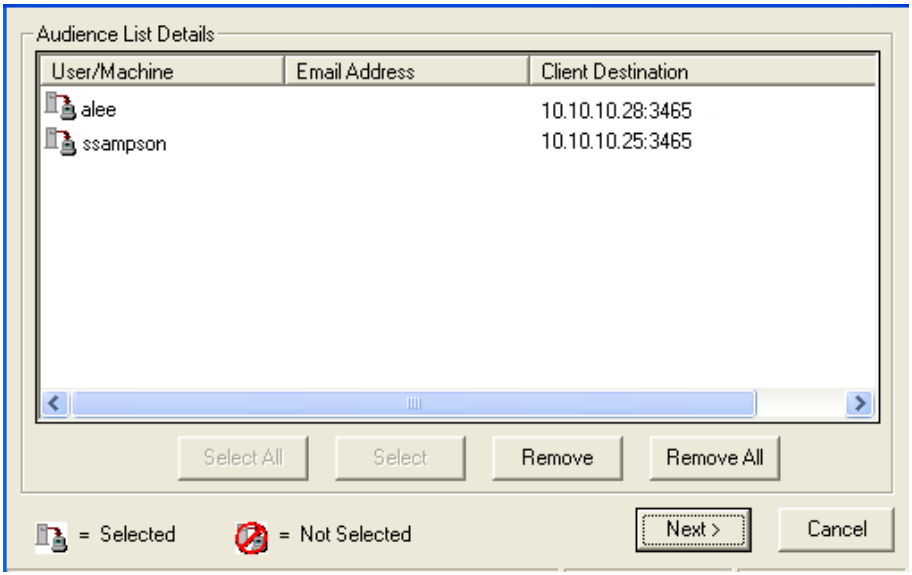
2 Select **Notify Subscribers**.

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 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 agent devices in the audience list and the number of agent devices selected are displayed at the bottom left of the dialog box.

4 Click **Next** when you are done selecting agent devices 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.

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 agent devices.

— **Remove the Application on the target machine(s)**
Select this option to remove an application from the agent devices.

Normally, Notify removes the application without requesting permission from the client. This allows removal of applications from unattended agent devices.

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.

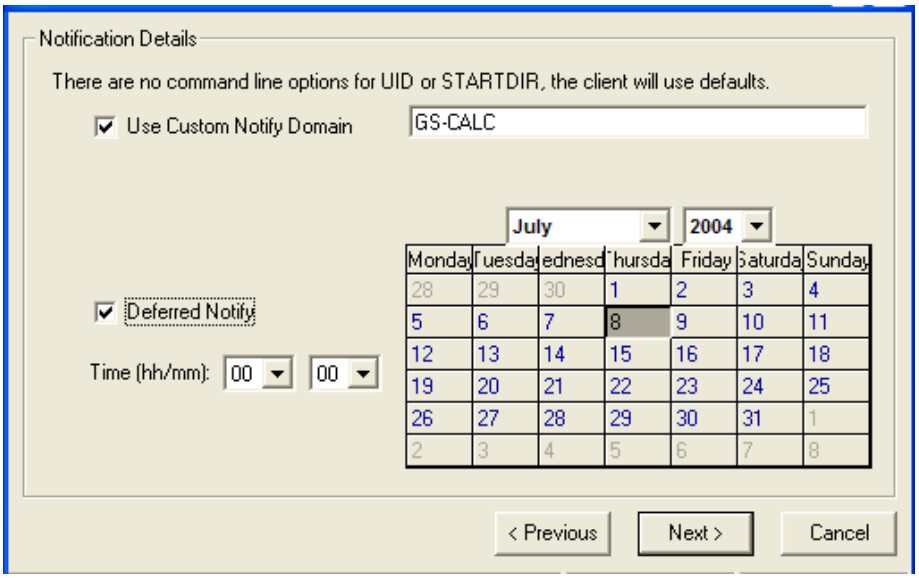
The Notification Details dialog box opens.

By default, the Notify occurs immediately and generates an object in the NOTIFY File in the CSDB. 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 CSDB.

— 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 agent device 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 agent device.

The Drag-and-Drop Notify works only if the agent device has connected to the Configuration Server prior to the notification. This populates the PROFILE File, which contains the agent device'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 agent devices.

To create a Drag-and-Drop Notify

1 Start the Admin CSDB Editor. The Admin CSDB Editor Security Information dialog box opens.

The User ID, as shipped from HP, is RAD_MAST. No password is necessary. This might have been changed during installation. 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 Admin CSDB Editor window opens.

3 Double-click **PRIMARY**.

4 Double-click **SYSTEM**.

5 Double-click **Application Manager (ZCOMMAND)**.

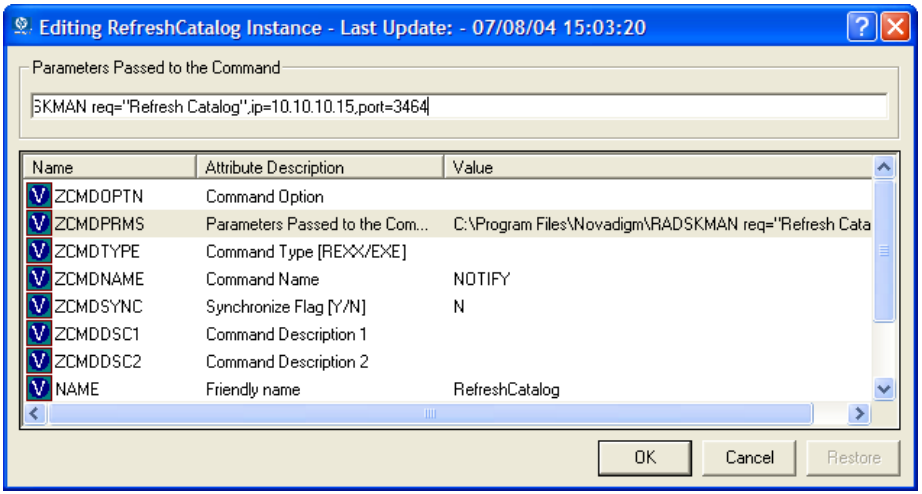
6 Right-click the **Mandatory** Instance, and select **Copy Instance**. The Copy Instance dialog box opens.

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 agent device. 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 186 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/HP/CM/Agent`).

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 agent device may be turned off.
- The subscriber does not have a valid e-mail address listed in the CSDB.

- The agent device is not running the Notify service.
- The agent device 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, HPCA 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 installation directory (by default, `/opt/HP/CM/ConfigurationServer/bin`).

```
[MGR_ATTACH_LIST]
ATTACH_LIST_SLOTS = 15
RESTART_LIMIT = 7
VERIFY_INTERVAL = 5
CMD_LINE=(zutilmgr) RESTART=YES
CMD_LINE=(zrexxmgr) RESTART=YES
CMD_LINE=(ztcpmgr PORT=3464,NAME=tcpmgr_3464) RESTART=YES
CMD_LINE=(znfytmgr NAME=NotifyManager) RESTART=YES
CMD_LINE=(zrtrymgr) RESTART=YES
```

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 *HP Client Automation Configuration Server User Guide (Configuration Server 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.

Viewing the Results of a Notify

You can find information about a Notify:

- In the Admin CSDB Editor in the NOTIFY File.
- In the Status Monitor accessed from the NOTIFY File in the Admin CSDB Editor.

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 (by default, `/opt/HP/CM/ConfigurationServer/log`).

Viewing an Instance in the NOTIFY File

The NOTIFY File is created after the first Notify is initiated. Each Notify operation creates a single object in the NOTIFY File in the CSDB. 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 Admin CSDB Editor, 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 40 Attributes in the NOTIFY instance

Attribute	Description
ZUSERID	The USER, WORKGRP, or DEPT that you notified.
ZCIPADDR	The IP address of the agent device.
EMAIL	The subscriber's e-mail address, if using e-mail notification.
NTFYTYPE	Indicates the type of notify, such as E for e-mail notification.
NTFYDATE	The date of the Notify.
NTFYTIME	The time of the Notify.
NTFYMSG	Message indicating the status of the Notify, such as "Successfully notified."
NTFYRC	The return code generated for a Notify.
NTFYCMDL	The command line that the Notify executed.
NTFYSUBJ	The subject of the email that is sent, if using e-mail notification.
LOCALUID	The user ID for the subscriber that is currently logged on to the computer.
NTFYRTIM	The time at which the notification should execute.
NTFYRNUM	The number of times to retry the Notify.
NTFYDOMN	The name of the domain where this instance is stored.
NTFYINS	The name of this instance.
NTFYPORT	The registered port for Notify.

Attribute	Description
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 agent device. 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 Admin CSDB Editor, 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.

Self Maintenance

Maintenance for the HPCA agents is available from Technical Support. The maintenance will include import decks for the CSDB. 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 agent's platform and current product level. Once you have decided to roll out the maintenance to the agent devices, you can add the service to the user's entitlements.

To minimize the need for separate PRDMAINT bundles based on different operating systems where the actual maintenance is the same, the ZMASTER.ZOSTYPE variables identify the operating system type or family. The valid values for this variable are:

- WIN32_NT (including Windows 2000, XP, and 2003)
- WINX64_NT
- WINIA64_NT
- UNIXLNUX
- MACOSX_PPC

Usage Notes

- All packages are disabled by default. This is accomplished by setting a ZSTOP expression to "1" to prevent deployment. Either remove this value for general deployment, or use this ZSTOP expression to restrict its deployment to certain groups.
- The first REQUIRES connection is reserved for any possible hot fix, a fix sent to you directly by Technical Support is not yet available in a fix or service pack. This package, _HOTFIX, will be used to chain any required fixes (and/or enhancements) and will be maintained by the customer. The second connection is for any locally customized code to be included as part of maintenance.
- Use the ACTMAINT attribute in the SETTINGS Class of the CLIENT Domain to specify how you want maintenance processed. You can choose to immediately download and install maintenance (I), download only and install later (D), or prompt users to install maintenance at another time (D). Maintenance runs *only* when the mnt parameter of radskman is set to Y. See ACTMAINT on page 108 and the mnt parameter on page 168 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 agent maintenance packages

1. A maintenance package is made available on the HP support 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 `bin` directory.
4. Import the files using the ZEDMAMS utility. For detailed information on the use of this utility, see 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, refer to 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 an agent connect process, the `mnt` parameter of the radskman command line, must be set to Y.

During catalog processing, the agent 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 HPCA 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 agents 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 HPCA subscriber computers.

For more information, refer to the *HP Client Automation Proxy Server Installation and Configuration Guide (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.

8 HPCA Application Self-service Manager User Interface

At the end of this chapter, you will:

- Know how to customize the HP Client Automation Application Self-service Manager (Application Self-service Manager).
- Understand how your subscribers can access the Application Self-service Manager user interface.
- Be able to use the Application Self-service Manager user interface, from a subscriber's perspective.

This guide describes the *suggested* implementation for the HP Client Automation (HPCA) Agents; Application Manager and Application Self-service 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 these agents This chapter covers customizing the agent and using the Application Self-service Manager user interface.

Subscribers use this interface to install, update, verify, or remove applications from their computers.

The term **application** refers to software or content that is available to the subscriber.

About the Application Self-service Manager User Interface

This section describes how to use the Application Self-service Manager user interface in detail. Although your subscribers will be using the user interface, you should be familiar with how it works.

Accessing the Application Self-service Manager User Interface

Before running the Application Self-service Manager, make sure your DISPLAY environment variable is set. See Table 3 on page 29 for more information.

To access the user interface for Linux

1. Change your current working directory to the directory where you installed the Application Self-service Manager (for Linux the default is`/opt/HP/CM/Agent/`).
2. Type **`./runrsm`**, and press **Enter**. The Subscriber Security Information dialog box opens.
3. If necessary, type your User ID and Password. If you do not know what these are, contact your network administrator.

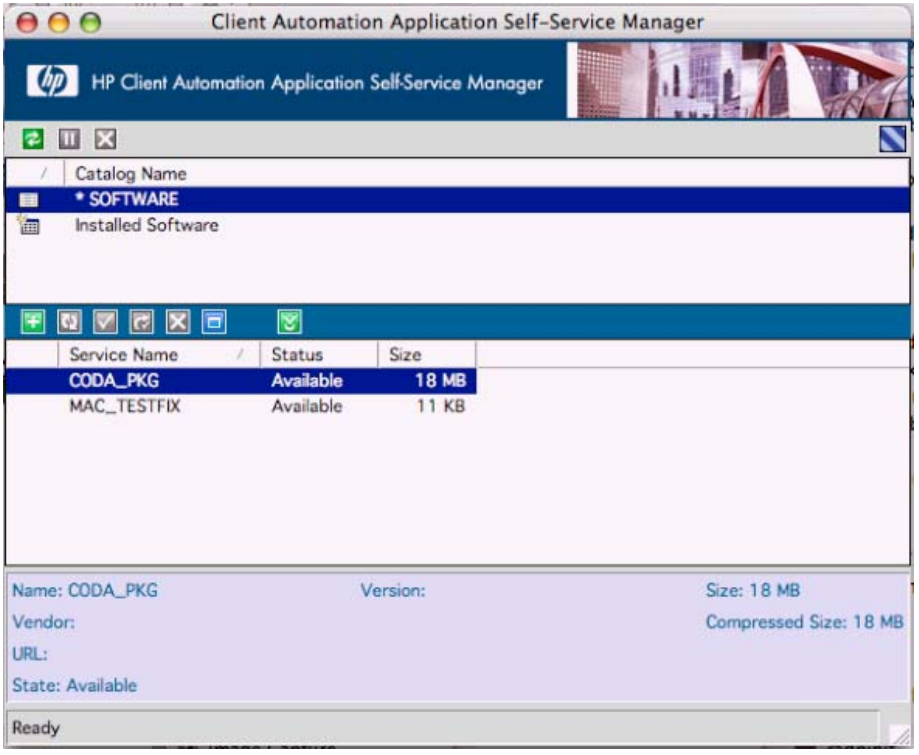
4 Click **OK**. The user interface opens. The user interface for Linux looks very similar to the user interface for Macintosh (see To access the user interface for Macintosh below).

To access the user interface for Macintosh

1 In the Finder navigate to where the Application Self-service Manager was installed (default location is `/Applications/HP/CM/Agent` or use the alias on the desktop named Application Self Service Manager. The Subscriber Security Information dialog box opens.

2 If necessary, type your User ID and Password. If you do not know what these are, contact your network administrator.

3 Click **OK**. The user interface opens.



Using the Application Self-service Manager User Interface

The Application Self-service Manager user interface has four main sections.

- **Global Toolbar**
  Allows you to refresh the catalog, pause the current action, or cancel the current action.
- **Catalog List**
  Lists the different software catalogs available.
- **Service List**
  Lists the applications that you are entitled to.
- **Application Self-service Manager menu options**
  Each section contains specific Application Self-service Manager options.

Global Toolbar

The Global Toolbar allows you to refresh the catalog, pause the current action, or cancel the current action. After an action has been paused, no other action can take place until you either resume the action, by clicking the Pause button again, or cancel the paused action by clicking the Cancel button.

Any time one of the buttons in the Global Toolbar is not available for the current action, they will appear grayed-out.

To refresh the catalog

- To refresh the selected catalog using the Global Toolbar, click **Refresh**.

To pause or resume the current action

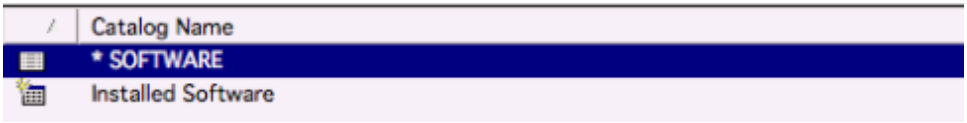
- To pause the current action using the Global Toolbar, click **Pause**.
- To resume a paused action, click **Resume**. (The Pause button is replaced with this button after you pause an action).

To cancel the current action

- To cancel the current action using the Global Toolbar, click **Cancel**.

Catalog Name List

The Catalog Name list section lists the available software catalogs and any virtual catalogs.



To select a catalog

- In the Catalog Name list, click on the Configuration Server catalog you would like to view in the Service List section. Refresh the catalog at any time by clicking the **Refresh** button in the Global Toolbar.

Virtual Catalogs

Virtual catalogs are subsets of the default catalog defined by specifying a name in the CATGROUP value for a service. Any services with the same CATGROUP value will be grouped together in a virtual catalog.

To set the CATGROUP attribute

The following example uses the Admin CSDB Editor, which is available for 32-bit Windows platforms.

1. Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.
2. If necessary, type a User ID and Password, and then click **OK**.

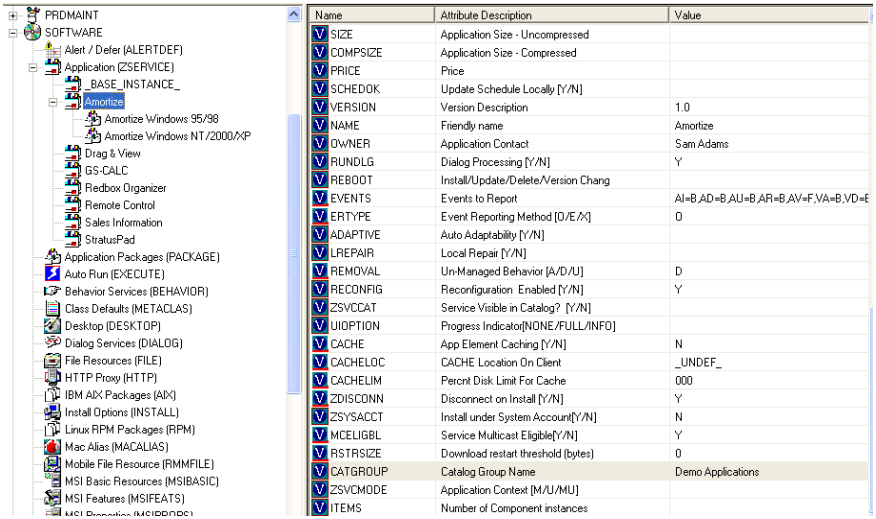
   The factory set user ID is RAD_MAST. No password is necessary. This might have changed during installation.

   You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

   The Admin CSDB Editor window opens.
3. Double-click **PRIMARY**.
4. Double-click **SOFTWARE**.
5. Double-click the name of the service you would like to add to a virtual catalog.

6 Double click the **CATGROUP** attribute and type the name of the virtual catalog you would like to add the service to.

7 Click **OK**.



The Service Name List

The Service Name list section lists the applications available to you. A check mark appears next to software that is already installed. The column headings displayed can be changed to suit your needs, see Preferences on page 214 for more information.

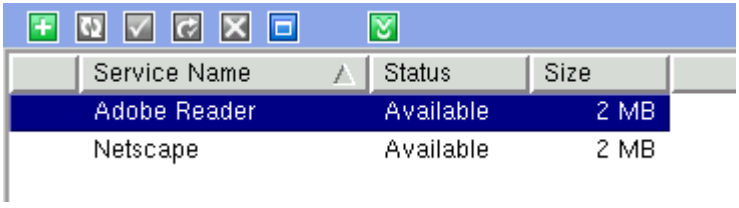


Table 41 Buttons in the Service Name list

Button	Action	Description
	Install	Installs the selected service on your machine
	Update	Updates the selected service.
	Verify	Verifies the files for the selected service.
	Repair	Repairs the selected service.

Button	Action	Description
	Remove	Removes the selected service from your machine.
	Expand/Collapse	Expands or collapses the selected service.
	Download Only	Download selected service from catalog into local cache without installing.

The buttons in the Service List section will be gray when they are not available for the selected application.

Some of the buttons in the Service List section are only available when the Show advanced operations box is checked in the Service List options section of the Preferences. See Preferences on page 214 for more information.

The Application Self-service Manager Menu Options

Use the Application Self-service Manager Menu options to configure and customize your Application Self-service Manager.

The following sections explain each option in the Application Self-service Manager menu in detail.

File

Use the **File** menu option to exit the Application Self-service Manager.

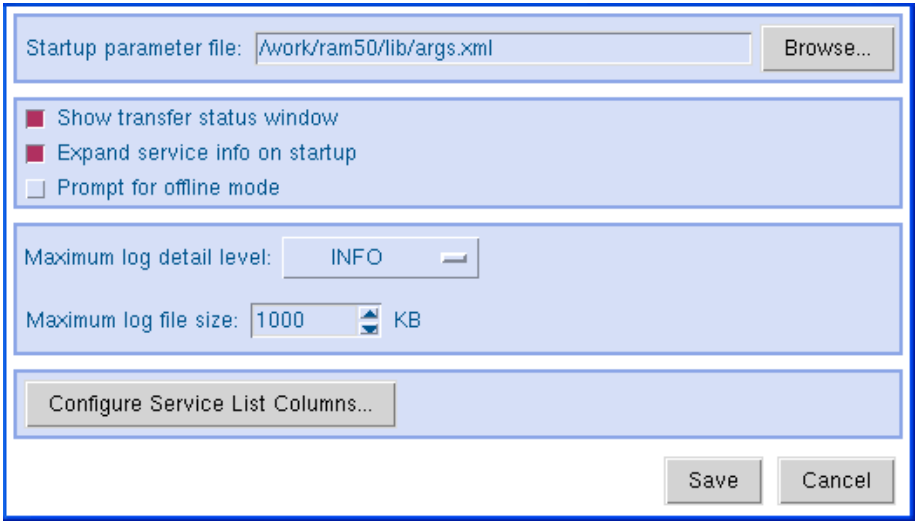
Action

Use the Action menu option to navigate between software catalogs, refresh the catalog, or view history or preference options.

- **Home**
  Click **Home** to return to the main Service List, displaying all available services in the Configuration Server Database.
- **My Software**
  Select this option to display only services that you have installed.
- **Refresh Catalog**
  Select this option to refresh the catalog, and check for updates to any available Services.

- **Preferences**
  Select this option to access various display options, service list options, and connection options for the Application Self-service Manager.

  At any point you can click on **Save**, or **Cancel** in the bottom right corner of the Preferences section to keep or disregard any changes you make.



  — **Startup parameter file**
  Enter the name and location of your Startup parameter file (by default: `/opt/HP/CM/Agent/lib/args.xml`). Click the **Browse** button to manually locate the file.

  — **Show transfer status window**
  Select this check box to display the transfer status window at the bottom of the Service List.

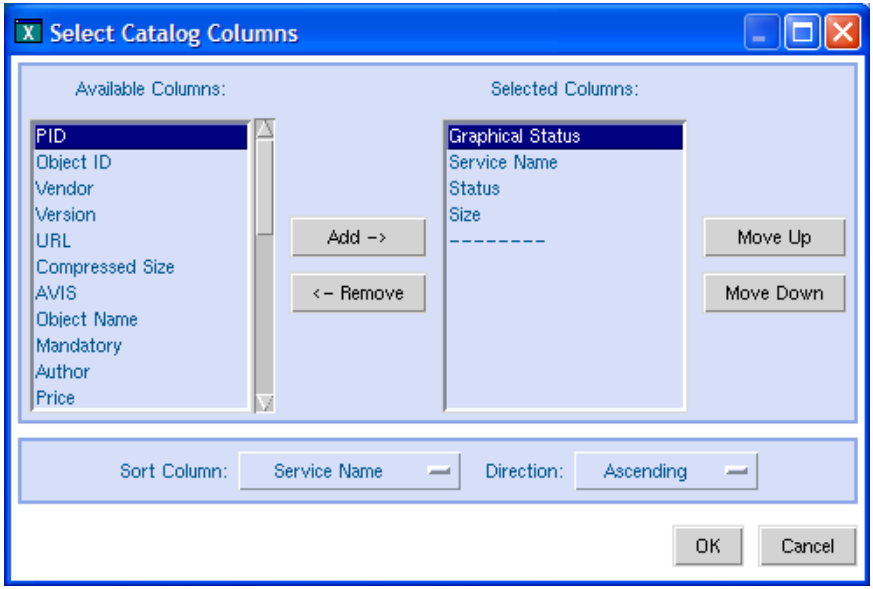
  — **Expand service info on startup**
  Select this check box to display the Service info at the bottom of the service list for the selected service.

  — **Prompt for offline mode**
  Select this check box if you would like to be prompted to work offline if a connection to the Configuration Server is not available.

  — **Maximum log detail level**
  Select the Log detail level you would like: ERROR, WARNING, INFO, DEBUG, or TRACE.

  — **Maximum log file size**
  Select the maximum log file size using the up and down arrows or by typing the log size into the text box.

— **Configure Service List Options**
Click the **Configure Service List Columns** button, as shown in the figure above, to modify the appearance of the Service List.



— **Customizing the Column Names in the Service List**

Use the columns area to customize the columns that appear in your service list. The right-hand column lists the column names currently displayed in your service list. For a description of each available column heading, see Table 42 on page 216.

To add columns to the Service List

1. In the Available Columns list box, select the column name you would like to add.
2. Click **Add**. The selected column is listed in the Selected Columns list box.
3. To change the order of the columns in the Selected Columns list, use the Move Up and Move Down buttons.
4. To set the default order of Services in the Service List, use the Sort Column and Direction buttons.
5. Click **OK** to return to the Preferences menu.
6. Click **Save** to keep your changes and return to the Application Self-service Manager.

To remove columns from the Service List

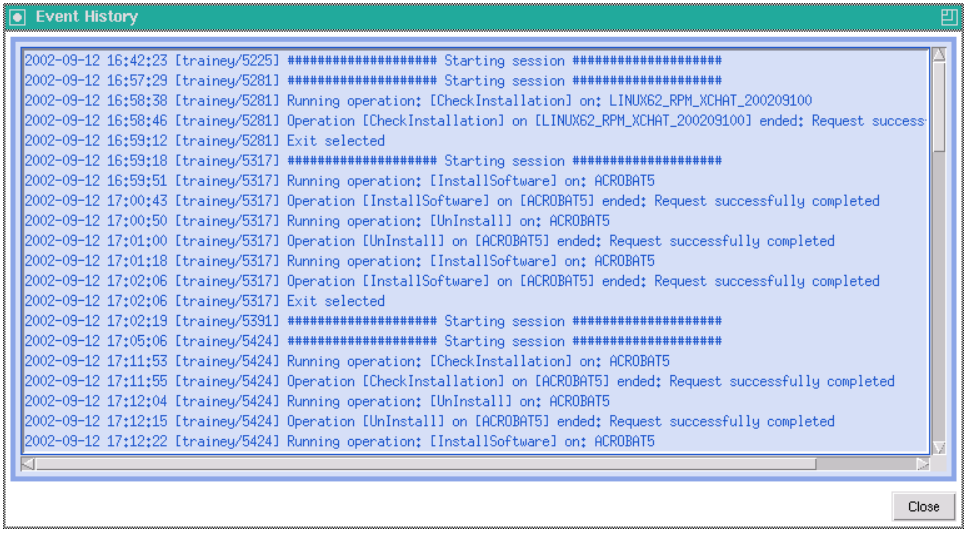
1. In the Selected Columns list box, select the column you would like to remove.
2. Click **Remove**. The selected column is removed from the Selected Columns list box and returned to Available Columns.
3. Click **OK** to return to the Preferences menu.
4. Click **Save** to keep your changes and return to the Application Self-service Manager.

Table 42 Column Headings Available for the Service List

Column Heading	Description
Author	The author of the service.
CompressedSize	The size of the compressed service (bytes).
Description	A short description of the service.
InstalledDate	The date the service was installed on your computer.
LocalRepair	If data is repairable locally (cached on your computer).
Mandatory	The Mandatory or Optional flags are displayed.
OwnerCatalog	The originating application domain name.
Price	Price of the service.
PublishedDate	The date the service was published to the catalog.
RepublishedDate	The date the service was republished to the catalog.
Service Name	Name of the Service (cannot be removed from the column display).
Size	The size of the service (bytes). Note: You need this amount of free space on your computer to successfully install the service.
Status	Current status of the software • Available • Installed • Update Available • Broken
UpgradedDate	The date the service was upgraded.

Column Heading	Description
Url	The software vendor's url.
Vendor	The software vendor who supplied the service.
VerifiedDate	The date the service was last verified.
Version	The version of the service.

— **History**
Select this option to display a history of the current session.



Services

The Services menu options are:

- Install
- Download
- Update
- Verify
- Repair
- Remove
- Information
- Schedule

Each Service option can be accessed by selecting a Service in the Service List and selecting the Service option from the Services menu. You can alternatively use the Service List buttons to perform these actions as well. The Service options are explained in detail in the following section.

Using the Application Self-service Manager User Interface

Installing Software

The applications that are available to you are listed in the Service list. You can install one or more of these applications at any time.

To install software

1. In the Service List, click the name of the software that you want to install.
2. Click **Install**.

   Some installations may display a set of dialog boxes. If so, follow the instructions. Otherwise, the installation begins immediately.

   A progress bar displays the installation progress.

   - Click **Cancel** in the Global Toolbar to cancel the installation.
   - Click **Pause** in the Global Toolbar to pause the installation. If you pause an action, you will not be able to perform any other actions until you either cancel or resume the currently paused action.

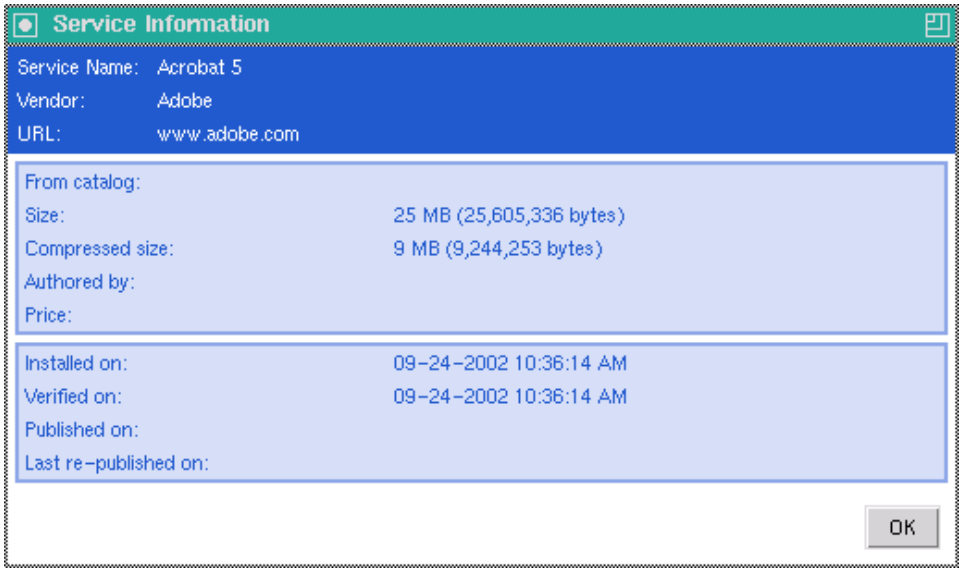
Refreshing the Catalog

The catalog is refreshed whenever you log on to the Application Self-service Manager user interface. While you are logged on, if you believe that the list of applications that you're authorized to use has changed, or that updates to your installed applications have become available, click Refresh Catalog in the Global Toolbar to retrieve the updated list of applications.

Viewing Information

You may want more information about an application than the Service List provides. If you would like to know the vendor, version, size, and date the application was installed, you can either add these columns to the Service List or double-click the selected service.

Click **OK** to close the Service Information window.



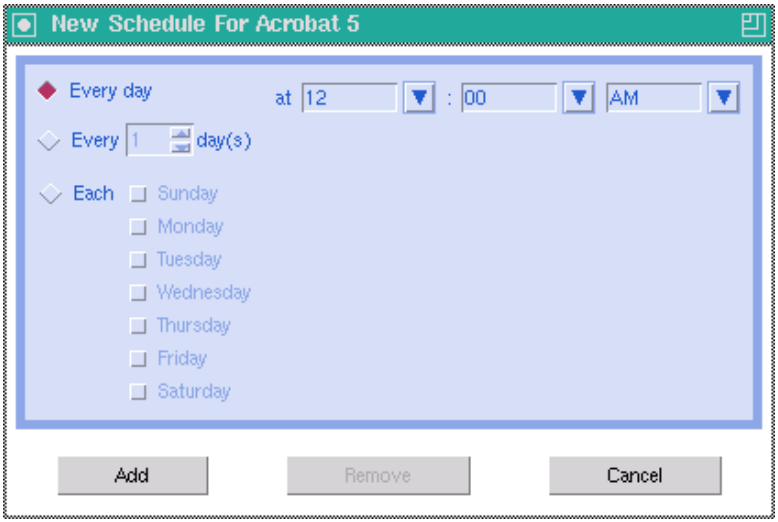
Scheduling Timed Events

After selecting an installed service, select Schedule from the Services menu to specify a schedule that will automatically update the applications that are installed on your computer. For example, you can schedule updates to occur during non-business hours, when you are not using your computer and network traffic is slower.

The Scheduling dialog box is only enabled when an Application Service (ZSERVICE) has the SCHEDOK attribute set to Y, indicating the Administrator authorized local scheduling capabilities on the selected service.

To schedule updates for an installed application

1. In the Application Self-service Manager user interface, select an installed application.
2. Select **Schedule** from the Services menu. The Scheduling dialog box opens.



3 Select one of the following:

— **Every day**
Updates occur every day at the specified time.

— **Every** n **days**
Updates occur every n days. Use the up and down arrows next to the **Every** option button to select the frequency of updates.

— **Each** weekday
Updates occur every weekday whose check box is selected. You may select more than one day.

4 Use the up and down arrows or type in the box labeled **at** to specify a specific time for the update.

5 Click **Add** to close the dialog box and accept the scheduled update.

Verifying Software

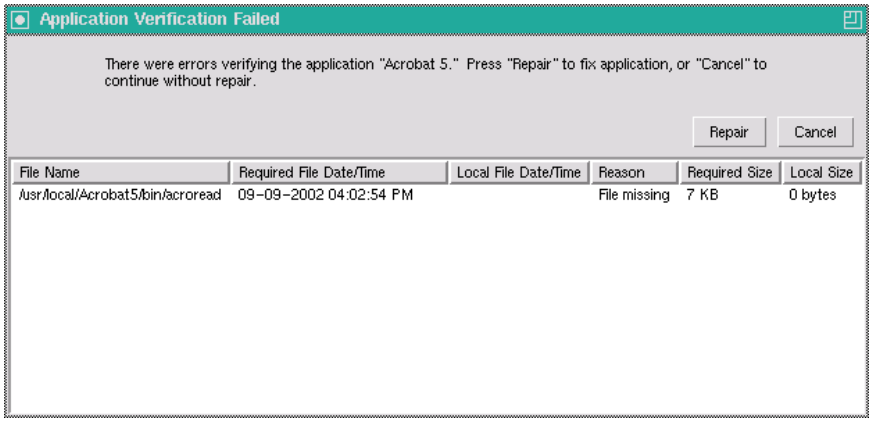
To check the installation of an application

1 In the Service List, select the installed service that you would like to verify.

2 Click **Verify**.

— If the application passes verification, the date and time of verification will appear in the Verified Date column for the application.

— If the application fails verification, a Verification Failed window will open displaying the problem with the application. Broken will appear in the Status column in the Service List.



3 To repair the software, either click **Repair** in the Verification Failed window, or click **Repair** in the Service List.

Repairing Software

If there is something wrong with an application, click **Repair** to fix it.

To repair software

1 Select an application that needs to be repaired (This is designated by an X in the first column).

2 Click **Repair**. HPCA retrieves the files needed to fix the application.

Removing Software

Use the **Remove** button to remove software from your computer.

To remove software

1 Select the software that you want to remove.

2 Click **Remove**.

3 Click **Yes** if you are asked to confirm that you want to remove the application.

Consolidated Agent Logs

In previous versions of HPCA, each of the main agent modules — radskman, radpinit, and radconct — created its own log, which would be overwritten each time the module ran. Use these logs to help you troubleshoot deployment problems. Beginning with HPCA 3.x, you can:

- Create a single log for all of these modules.
- Append information to the log if you prefer to see all of the activity.
- Name the log, which may be useful for debugging deployments or for collecting information from your agent devices. 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.

Each of the three main agent modules takes command line parameters in the following format:

```
Keyword = value (in comma-delimited format)
```

Therefore, you can use the following optional parameters on the command line if you want to name the log file or append information to an existing file. For example, you could add the log parameter to a radskman command line in a Notify to specify a particular log name.

Table 43 Parameters for Log Files

Parameter	Description	Default	Example
Alog	The name of the log file to append to. If you do not specify the alog parameter, the log specified in the log parameter will be appended to. Use a valid file name without a path. By default logs are stored in the `IDMLOG` folder.	N/A	`Alog=Application1.log`

Parameter	Description	Default	Example
Log	The name of the log file to create. Use a valid file name without a path. By default logs are stored in the `IDMLOG` folder. If there is an existing log file with the same name, HPCA creates a backup of that file called *logname*.bak. If there is already a *logname*.bak, HPCA will overwrite it.	`Connect.log`	`Log=20010524.log`

The value for the log parameter is stored in the LOGNAME attribute, located in the ZMASTER object in the catalog and application directories.

Self Maintenance

Maintenance for the agents is available from Technical Support. The maintenance will include import decks for the Configuration Server 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 agent's platform and current product level. After you have decided to roll out the maintenance to the agent devices, you can add the service to the user's entitlements.

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 108 for more information. For details on radskman, refer to the previous chapter, Deploying Services.

We will provide an updated PRDMAINT instance with each new maintenance pack. The customer is not required to apply all maintenance.

To deploy agent maintenance packages

1. A maintenance package is made available on the HP 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, refer to the *Configuration Server Guide.*
5. Restart the Configuration Server.
6. Assign the Maintenance Server to the appropriate users in the POLICY Domain.

   To run the maintenance portion of an agent connect process, the mnt parameter of the radskman command line, must be set to Y.

During catalog processing, the agent 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.

Application Deferrals

When an application is deployed, an administrator can display a deferral message based on the network threshold, the data download size, a specified date setting, or a deferral count. When an application has data that needs to be downloaded to the agent device, the agent will check if the application is configured for deferral. If it is, the agent will check the current bandwidth setting against the administrator specified bandwidth threshold setting. If the current network speed is less than the Network Threshold (DT) value, a deferral message will be displayed asking the subscriber if he wants to defer the deployment.

The administrator can configure the number of times an application can be deferred, the date an application can be deferred until, or a minimum byte count to alert on. If the number of deferrals or the deferral date has been reached, the application will be installed or updated without displaying a deferral message. If the size of the data is less than the minimum byte count, the alert panel will be skipped.

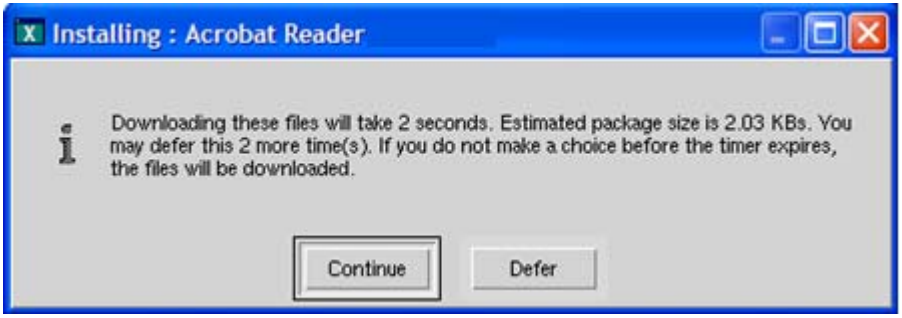
If the application has been configured for a deferral, and all of the requirements listed below are met, the agent will display the deferral dialog box.

- The Alert Mode (DM) is configured for the current operation: Install, Update, or Both.
- The current network speed is lower than Network Threshold Speed (DT).
- The UIOPTION attribute in the ZSERVICE instance is not set to NONE.
- The data to be downloaded is greater than the administrator specified minimum byte count, (DBT) and lower than the specified maximum byte count (DAT).
- If specified, the deferral date, Allow Install Deferral up to (DI), or Allow Update Deferral up to (DU) has been reached.

  or

- The number of deferrals allowed (DN) has been reached.

If these requirements are met, and you are using the Application Self-service Manager, you will be prompted to continue or defer.



The subscriber can choose to defer the action or to continue with it.

If the timeout value is exceeded, the action will be taken that is identified in the DA (Action on Timeout Cont/Defer) attribute.

To implement an Application Deferral, you will need to create an instance in the Alert/Defer (ALERTDEF) Class, and connect that instance to the appropriate Application (ZSERVICE) instance.

Creating a Deferral Instance

The Alert/Defer (ALERTDEF) class has been added to the SOFTWARE Domain in the Configuration Server Database to configure application alerts. In order to configure an alert, you will need to create an instance in the Alert/Defer (ALERTDEF) Class.

To create an instance of the Alert/Defer (ALERTDEF) Class

1 Go to **Start** → **Programs** → **HP Client Automation Administrator** → **HP Client Automation Administrator CSDB Editor**. The Admin CSDB Editor Security Information dialog box opens.

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

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

You can change this by selecting the Change Password check box and typing the new password in the New Password and Verify New Password text boxes.

8 The Admin CSDB Editor window opens.

9 Double-click **PRIMARY**.

10 Double-click **SOFTWARE**.

11 Right-click **Alert/Defer (ALERTDEF)**. A shortcut menu opens.

12 Click **New Instance**. The Create Instance dialog box opens.

13 Type in a name for the new instance. In our example, we create an instance called SalesDefer.

14 Click **OK**.

The new instance is created.

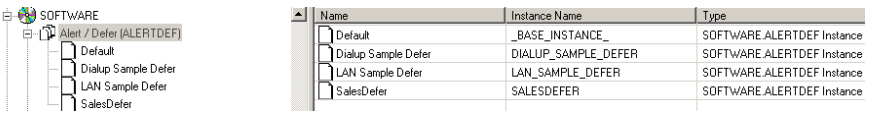
Configuring a Deferral

After the instance is created, it must be configured. The Alert/Deferral (ALERTDEF) class includes two sample instances, Dial Up Sample Defer, and LAN Sample Defer.

To configure an Alert/Deferral (ALERTDEF) instance

1 Use the Admin CSDB Editor to navigate to the Alert/Defer (ALERTDEF) instance you want to edit.

2 Double-click the instance. In this example, we are editing the SalesDefer instance.



3 Double-click the variable you want to edit. See Table 44 below for information on the attributes for this class.

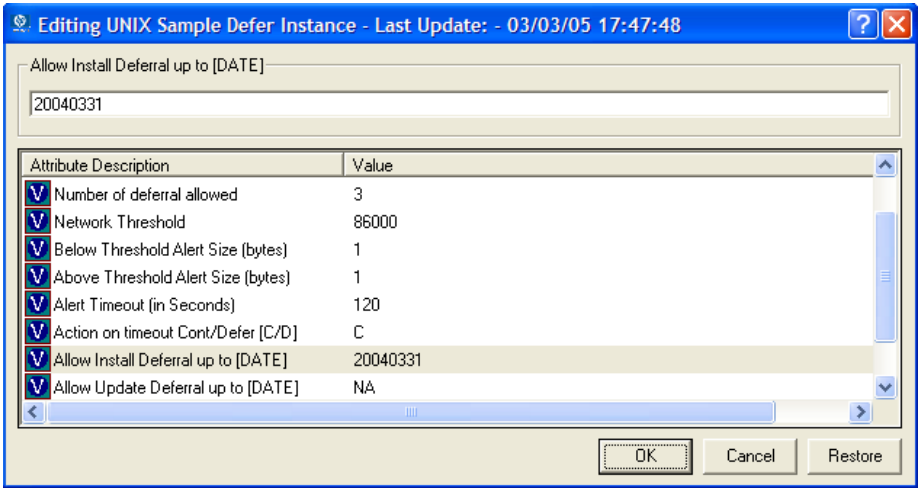
Table 44 Attributes in the ALERTDEF Class

Attribute	Description	Default Value
ALERTMSG	**Alert Message** An exclamation point (!) preceding "Service Alert Message" denotes a high priority message.	
DM	**Alert Mode [I/U/B]** • Set to I for Install. • Set to U for Update. • Set to B for either Install or Update.	B

Attribute	Description	Default Value
DN	**Number of deferrals allowed** Maximum number of deferrals allowed before the action specified in DA (Deferral Action) is taken.	0
DT	**Network Threshold** Network bandwidth threshold in bytes. The current network speed must be less than this value to meet the deferral requirement.	86000
DBT	**Below Threshold Alert Size (bytes)** The size of the file to be downloaded must be greater than this value to meet the deferral requirement.	50000
DAT	**Above Threshold Alert Size (bytes)** The size of the file to be downloaded must be less than this value to meet the deferral requirement. If the value is 0, this attribute is ignored.	0
DTO	**Alert Timeout (in seconds)** Specify in seconds the amount of time to display the Defer Alert dialog box. After the timeout is reached, the action specified in the DA (Action on timeout) variable, will be taken.	120
DA	**Action on timeout Cont/Defer [C/D]** Specify **C** to continue with the specified action if the subscriber does not respond to the defer alert dialog box by the value specified in the DTO (Alert Timeout). Specify D to defer the specified action.	C
DI	**Allow Install Deferral up to [DATE]** After this date is reached, the option to defer installation will no longer be available. The application will be installed. This should be in the format YYYYMMDD.	NA
DU	**Allow Update Deferral up to [DATE]** After this date is reached, the option to defer an application update will no longer be available. The application will be updated. This should be in the format YYYYMMDD.	NA

Attribute	Description	Default Value
Name	**Friendly Name** Friendly Name for the instance.	Default
DEFOPTNS	**Defer Options (Do not Modify)** This attribute is used to resolve the values of the other attributes of this class. *Do not modify*.	&(DM),&(DN),&(DT),&(DBT),&(DAT),&(DTO),&(DA),&(DI),&(DU)

In this example, we want to add an install deferral date. To do this, double-click the DI variable in the list view.



4 Enter the date up to which you will allow the application installation to be deferred.

5 Click on the next attribute, and type in the appropriate value.

6 Click **OK** when you are finished editing the attributes. The Instance Edit Confirmation dialog box opens.

7 Click **Yes** to confirm the changes.

The changes are made to the Alert/Defer (ALERTDEF) instance.

After the Alert/Defer (ALERTDEF) instance is created, you need to connect the Alert/Defer (ALERTDEF) instance to an Application (ZSERVICE) instance. To do this, use the Admin CSDB Editor to click and drag the Alert/Defer (ALERTDEF) instance to the appropriate Application (ZSERVICE) instance. For additional information on using the Admin CSDB Editor, see the *Administrator User Guide*.

Now that you are familiar with how your subscribers will manage their software, you may want to see how you can gather information about the agent device, the subscriber, or the results of a subscriber's activity. See, Chapter 3, HPCA Agent Directories and Objects for more information.

Summary

- Subscribers use the Application Self-service Manager user interface to install, update, verify, and remove software on their computers.
- Customize the user interface using the Preferences action menu option.

Glossary

Admin Agent Explorer

The Admin Agent Explorer (Object Editor) can be used to view or edit local objects, or create new objects. You can also use the Admin Agent Explorer to view objects located on a file server or on other computers to which you are connected via a local area network (LAN).

Admin CSDB Editor

The Admin CSDB Editor is used to manipulate the contents of the Configuration Server Database.

Admin Publisher

The Admin Publisher is used to create packages of data and store them in (i.e., promote them to) the Configuration Server Database.

Administrative Installation Point (AIP)

An AIP is a server share or local directory structure that contains all of the files needed to run setup for a Windows Installer-enabled application.

agent

The agent runs on the user's computer. It communicates with the Configuration Server to receive information about the desired state of the subscriber's computer, and compares that information to the actual state of the subscriber's computer. Then, the agent makes any adjustments necessary to make the actual state match the desired state.

agent device

An agent device is a user's computer that has the agent software installed on it.

agent object

An agent object is a file located on the agent device that contains information about the configuration of services or hardware.

APPEVENT

APPEVENT is the agent object that provides information about an application event, such as success or failure of the installation.

Application Manager

The Application Manager Radskman is the agent executable that manages mandatory services. The administrator uses the Admin CSDB Editor to specify the services that the Application Manager manages on the subscriber's computer. No user interface is available.

Application Self-service Manager

The Application Self-service Manager is the agent used to manage optional services. The administrator uses the Admin CSDB Editor to specify the services that are available to the subscriber.

The subscriber installs and manages data that is available from the Application Self-service Manager user interface (Service List).

applications

Also called software, data, or services.

Applications are one type of content that HPCA can manage on subscriber computers. Use the Admin Publisher to create packages of data to be managed on your subscribers' computers.

attribute

Also called *field, variable,* 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 agent device. Each attribute defined in a class template has a set of Configuration Server properties and a set of agent properties.

audience list

An audience list is a directory of the subscribers for an application used by 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 Configuration Server Database against the resources currently being published.

class

A class defines a category of the distribution model to be managed. It is conceptually similar to a schema in a relational database structure or a file layout in a traditional flat file. Each of the required elements of a distribution model (e.g., users, applications, etc.) is defined in the Configuration Server Database by its class.

class connection variable

A class connection variable determines the path of resolution for an agent's distribution model during the agent 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

Also called *instance*.

A class instance is an object in the Configuration Server Database that contains 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.

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

See agent.

Client Explorer

See Admin Agent Explorer.

Configuration Server

Also called *Active Component Server* or *Manager*.

The Configuration Server distributes applications to agent devices. It runs on the server and maintains the Configuration Server Database, which stores information that the Configuration Server needs to manage data for distribution to agent devices.

Configuration Server Database

The Configuration Server Database (CSDB) stores all of the information necessary to manage data on an agent device, including:

- The software and/or data that HPCA distributes.
- The desired state of each agent device with respect to the HPCA - managed content.
- The policies determining which subscribers can subscribe to which packages.
- Security and access rules for HPC administrators.

Use the Admin CSDB Editor to manipulate the Configuration Server Database.

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 Server class instance. Typically, this class' instances have distributable data associated with them such as FILE, REGISTRY, or DESKTOP.

Use the Admin CSDB Editor'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 Admin CSDB Editor's Class Editor to set the class type to "Configuration".

Configuration Server

See Configuration Server.

Database

See Configuration Server Database.

desired state

The desired state embodies the content that HPCA manages for a specific subscriber's computer. A model representing the desired state for each subscriber's computer is stored in the Configuration Server Database. The desired state model is created and managed using the Admin CSDB Editor.

domain

A domain logically partitions a file in the Configuration Server 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. HPCA 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 Configuration Server 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 HPCA. Others are the NOTIFY file and the PROFILE File. HPCA administrators will do most of their work in the PRIMARY File.

instance

Also called *class instance*.

An instance is a Configuration Server 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

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 Inventory Manager agent is a WBEM (Web-based Enterprise Management) consumer.

mandatory service

A mandatory service is a service that is required on the subscriber's 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 HP-supplied REXX run-time environment interprets REXX methods.

Agent methods run on the subscriber's 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 subdirectory for .exe methods or in the Manager REXX subdirectory for REXX methods.

For agent methods, it contains the name of the method to execute on the subscriber's computer. The name of a method variable that executes an agent method identifies the event (such as installing or removing software) for which the method should be executed. Agent methods are located in the `IDMSYS` location on the subscriber's computer.

Notify

A notify forces one or more agent devices to connect to the Configuration Server to 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 is attempted to a non-existent instance of a class, 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` suffix on the agent device. 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 Admin Agent Explorer to view, edit, or create objects.

optional service

An optional service is a service that is available to subscribers via the Service List of the Application Self-service 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 AdminPublisher, you are storing the package in the Configuration Server Database.

publish

To bundle a set of related data into a single unit that can be managed by HPCA.

Publisher

See Admin Publisher.

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 Configuration Server Database and parameters included in the service request itself.

In other words, what HPCA does depends upon what information is stored in the Configuration Server Database and what information accompanies the request for HPCA to perform some action.

For example, the agent 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 *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

The Scheduler service (radsched), installed with the Application Manager, allows you to deploy a service at a specific time.

service

Also called a software application, application, or software.

A service is a group of related packages.

session

A session identifies a packaging exercise in Admin Publisher that results in the creation of one package.

Software Manager

See Application Self-service Manager.

symbol

A symbol is the name of a variable in global memory, preceded by an ampersand.

symbolic substitution

Database instances and agent 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, HPCA 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 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 HPCA logon dialog box on the agent, when the subscriber visits the 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 Configuration Server Database doesn't change. Only the value in the in-storage object derived from the Configuration Server 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 Admin CSDB Editor.

Timer

See Scheduler.

variable

A variable is a piece of named storage that contains a changing value. The variable's value forms a part of the agent'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 HPCA 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 pre-determined 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 agent device such as processor, operating system, and drives.

ZMASTER

The ZMASTER object contains information about the agent device that is necessary to run the Application Manager such as the identity of the subscriber and the IP address of the agent device.

ZTIMEQ

The ZTIMEQ object is created, based on information in the Scheduler (TIMER) instance, when a timer is deployed to the agent device.

Index

B

C

D

E

F

G

H

I

J

L

M

N

O

P

Q

R

S

T

U

V

W

Z